ALGORITMOS

Contenido

INTRODUCCION 5

CAPITULO I. 6

CONCEPTOS BÁSICOS Y METODOLOGÍA PARA LA SOLUCIÓN DE PROBLEMAS POR MEDIO DE COMPUTADORAS. 6

1.1 Introducción 7

1.2 Definición de Lenguaje 10

1.3 Definición de Algoritmo 11

1.4 Tipos de Algoritmos 11

1.5 Lenguajes Algorítmicos 12

1.6 Metodología para la solución de problemas por medio de computadora 12

1.6.1. Definición del Problema 12

1.6.2 Análisis del Problema 12

1.6.3 Diseño del Algoritmo 13

1.6.4 Codificación 13

1.6.5 Prueba y Depuración 13

1.6.6 Documentación 14

1.6.7 Mantenimiento 15

CAPITULO II. 16

ENTIDADES PRIMITIVAS PARA EL DESARROLLO DE ALGORITMOS 16

2.1 Tipos De Datos 16

2.2 Expresiones 18

2.3 Operadores y Operandos 18

2.4 Identificadores 25

CAPITULO III. 28

TÉCNICAS DE DISEÑO 28

3.1 Top Down 28

3.2 Bottom Up 29

CAPITULO IV. 29

TÉCNICAS PARA LA FÓRMULACIÓN DE ALGORITMOS 29

4.1 Diagrama de Flujo 30
4.2 Pseudocódigo 32
4.3 Diagramas estructurados (Nassi-Schneiderman) 32
CAPITULO V. 34
ESTRUCTURAS ALGORITMICAS 34
5.1. Estructuras Secuenciales 35
5.1.1 Problemas Secuenciales 36
5.1.2 Problemas Propuestos Secuenciales 39
5.2 Estructuras de Condicionales 41
5.2.1 Problemas Condicionales 43
5.2.2 Problemas Propuestos 59
5.2.3 Condicionales Múltiples 62
5.3. Estructuras Cíclicas 64
5.3.1 Hacer-Para 64
5.3.2 Hacer-Mientras 73
5.3.3 Repetir-Hasta 77
5.3.3.2 Problemas Repetitivos Compuestos 81
VI ARREGLOS 84
6.1 Arreglos Unidimensionales 86
6.1.1 PROBLEMAS PROPUESTOS 90
6.2. Arreglo Bidimensional 96
6.2.1 PROBLEMAS PROPUESTOS 97
VII FUNCIONES 102
7.1 Módulo 102
7.1.1 Ventajas de la Programación Modular 103
7.1.2 Desarrollar programas de forma modular 103
7.1.3 Tiempo de vida de los datos – Ámbito. 104
7.1.4 PARÁMETROS 105
7.1.5 FUNCIONES 107
7.1.6 PROBLEMAS PROPUESTOS 108
VIII METODOS DE ORDENAMIENTO 108
8. 1 ORDENAMIENTOS INTERNOS Y EXTERNOS 108

INTRODUCCION

El desarrollo de algoritmos es un tema fundamental en el diseño de programas por lo cual el alumno debe tener buenas bases que le sirvan para poder desarrollar de manera fácil y rápida sus programas.

Estos apuntes servirán de apoyo al docente del SENA, en su labor cotidiana de enseñanza y al estudiante le facilitará desarrollar su capacidad analítica y creadora, para de esta manera mejorar su destreza en la elaboración de algoritmos que sirven como base para la codificación de los diferentes programas que tendrá que desarrollar a lo largo de su carrera.

CAPITULO I.

CONCEPTOS BÁSICOS Y METODOLOGÍA PARA LA SOLUCIÓN DE PROBLEMAS POR MEDIO DE COMPUTADORAS.

1.1 Introducción

- De los problemas a los programas
- Breves prácticas de programación

1.2 Definición de lenguaje

1.3 Definición de algoritmo

1.4 Algoritmos cotidianos

1.5 Definición de lenguajes algorítmicos

1.6 Metodología para la solución de problemas por medio de computadora

1.7 Definición del problema

1.8 Análisis del problema

1.9 Diseño del algoritmo

1.10 Codificación

1.11 Prueba y depuración

1.12 Documentación

1.13 Mantenimiento

OBJETIVO EDUCACIONAL:

El alumno:

- Conocerá la terminología relacionada con los algoritmos; así como la importancia de aplicar técnicas adecuadas de programación.
- Conocerá la metodología en cada una de sus etapas.

1.1 Introducción

La computadora no solamente es una máquina que puede realizar procesos para darnos resultados, sin que tengamos la noción exacta de las operaciones que realiza para llegar a esos resultados. Con la computadora además de lo anterior también podemos diseñar soluciones a la medida, de problemas específicos que se nos presenten. Más aún, si estos involucran operaciones matemáticas complejas y/o repetitivas, o requieren del manejo de un volumen muy grande de datos.

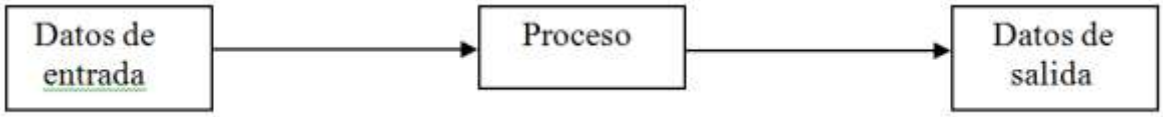
El diseño de soluciones a la medida de nuestros problemas, requiere como en otras disciplinas una metodología que nos enseñe de manera gradual, la forma de llegar a estas soluciones.

A las soluciones creadas por computadora se les conoce como programas y no son más que una serie de operaciones que realiza la computadora para llegar a un resultado, con un grupo de datos específicos. Lo anterior nos lleva al razonamiento de que un programa nos sirve para solucionar un problema específico.

Para poder realizar programas, además de conocer la metodología mencionada, también debemos de conocer, de manera específica las funciones que pueden realizar la computadora y las formas en que se pueden manejar los elementos que hay en la misma.

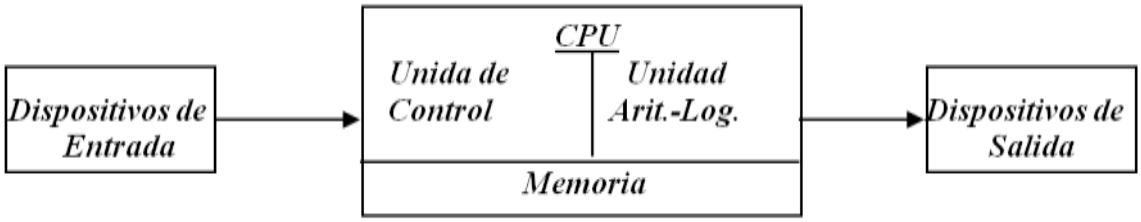
Computadora: Es un dispositivo electrónico utilizado para procesar información y obtener resultados. Los datos y la información se pueden introducir en la computadora como entrada (input) y a continuación se procesan para producir una salida (output).

Proceso de información en la computadora



Programa: Es el conjunto de instrucciones escritas de algún lenguaje de programación y que ejecutadas secuencialmente resuelven un problema específico.

Organización física de una computadora



Dispositivos de Entrada: Como su nombre lo indica, sirven para introducir datos (información) en la computadora para su proceso. Los datos se leen de los dispositivos de entrada y se almacenan en la memoria central o interna. Ejemplos: teclado, scanners (digitalizadores de rastreo), mouse (ratón), trackball (bola de ratón estacionario), joystick (palancas de juego), lápiz óptico.

Dispositivos de Salida: Regresan los datos procesados que sirven de información al usuario. Ejemplo: monitor, impresora.

La Unidad Central de Procesamiento (C.P.U) se divide en dos:

- Unidad de control
- Unidad Aritmético - Lógica

Unidad de Control: Coordina las actividades de la computadora y determina que operaciones se deben realizar y en qué orden; así mismo controla todo el proceso de la computadora.

Unidad Aritmética - Lógica: Realiza operaciones aritméticas y lógicas, tales como suma, resta, multiplicación, división y comparaciones.

La Memoria de la computadora se divide en dos:

- Memoria Central o Interna
- Memoria Auxiliar o Externa

Memoria Central (interna): La CPU utiliza la memoria de la computadora para guardar información mientras trabaja con ella; mientras esta información permanezca en memoria, la computadora puede tener acceso a ella en forma directa. Esta memoria construida internamente se llama memoria de acceso aleatorio (RAM).

La memoria interna consta de dos áreas de memoria:

La memoria RAM (Randon Access Memory): Recibe el nombre de memoria principal o memoria del usuario, en ella se almacena información solo mientras la computadora esta encendida. Cuando se apaga o arranca nuevamente la computadora, la información se pierde, por lo que se dice que la memoria RAM es una memoria volátil.

La memoria ROM (Read Only Memory): Es una memoria estática que no puede cambiar, la computadora puede leer los datos almacenados en la memoria ROM, pero no se pueden introducir datos en ella, o cambiar los datos que ahí se encuentran; por lo que se dice que esta memoria es de solo lectura. Los datos de la memoria ROM están grabados en forma permanente y son introducidos por el fabricante de la computadora.

Memoria Auxiliar (Externa): Es donde se almacenan todos los programas o datos que el usuario desee. Los dispositivos de almacenamiento o memorias auxiliares (externas o secundarias) más comúnmente utilizados son: cintas magnéticas y discos magnéticos.

1.2 Definición de Lenguaje

Lenguaje: Es una serie de símbolos que sirven para transmitir uno o más mensajes (ideas) entre dos entidades diferentes. A la transmisión de mensajes se le conoce comúnmente como comunicación.

La comunicación es un proceso complejo que requiere una serie de reglas simples, pero indispensables para poderse llevar a cabo. Las dos principales son las siguientes:

- Los mensajes deben correr en un sentido a la vez.
- Debe forzosamente existir 4 elementos: Emisor, Receptor, Medio de Comunicación y Mensaje.

Lenguajes de Programación

Es un conjunto de símbolos, caracteres y reglas (programas) que les permiten a las personas comunicarse con la computadora.

Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, calculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación.

Los lenguajes de programación se clasifican en:

- **Lenguaje Maquina:** Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje maquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 o 1).
- **Lenguaje de Bajo Nivel (Ensamblador):** En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- **Lenguaje de Alto Nivel:** Los lenguajes de programación de alto nivel (BASIC, pascal, cobol, fortran, java, php, javascript, c, c++, c#, f++, etc.) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos (en general en inglés), lo que facilita la escritura y comprensión del programa.

1.3 Definición de Algoritmo

La palabra algoritmo se deriva de la traducción al latín de la palabra árabe alkhowarizmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

1.4 Tipos de Algoritmos

- **Cualitativos**: Son aquellos en los que se describen los pasos utilizando palabras.
- **Cuantitativos**: Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.

1.5 Lenguajes Algorítmicos

Es una serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso.

Tipos de Lenguajes Algorítmicos

- **Gráficos**: Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).
- **No Gráficos**: Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

1.6 Metodología para la solución de problemas por medio de computadora

1.6.1. Definición del Problema

Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.

1.6.2 Análisis del Problema

Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:

Los datos de entrada.

Cuál es la información que se desea producir (salida)

Los métodos y fórmulas que se necesitan para procesar los datos.

Una recomendación muy práctica es el que nos pongamos en el lugar de la computadora y analicemos que es lo que necesitamos que nos ordenen y en que secuencia para producir los resultados esperados.

1.6.3 Diseño del Algoritmo

Las características de un buen algoritmo son:

- Debe tener un punto particular de inicio.
- Debe ser definido, no debe permitir dobles interpretaciones.
- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.
- Debe ser finito en tamaño y tiempo de ejecución.

1.6.4 Codificación

La codificación es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudocódigo), en una serie de instrucciones detalladas, en un código reconocible por la computadora, la serie de instrucciones detalladas se le conoce como código fuente, el cual se escribe en un lenguaje de programación o lenguaje de alto nivel.

1.6.5 Prueba y Depuración

Los errores humanos dentro de la programación de computadoras son muchos y aumentan considerablemente con la complejidad del problema. El proceso de identificar y eliminar errores, para dar paso a una solución sin errores se le llama depuración.

La depuración o prueba resulta una tarea tan creativa como el mismo desarrollo de la solución, por ello se debe considerar con el mismo interés y entusiasmo.

Resulta conveniente observar los siguientes principios al realizar una depuración, ya que de este trabajo depende el éxito de nuestra solución.

1.6.6 Documentación

Es la guía o comunicación escrita es sus variadas formas, ya sea en enunciados, procedimientos, dibujos o Diagramas.

A menudo un programa escrito por una persona, es usado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).

La documentación se divide en tres partes:

Documentación Interna

Documentación Externa

Manual del Usuario

- **Documentación Interna**: Son los comentarios o mensaje que se añaden al código fuente para hacer más claro el entendimiento de un proceso.
- **Documentación Externa**: Se define en un documento escrito los siguientes puntos:

  Descripción del Problema

  Nombre del Autor

  Algoritmo (diagrama de flujo o pseudocódigo)

Diccionario de Datos

Código Fuente (programa)

- **Manual del Usuario**: Describe paso a paso la manera como funciona el programa, con el fin de que el usuario obtenga el resultado deseado.

1.6.7 Mantenimiento

Se lleva acabo después de terminado el programa, cuando se detecta que es necesario hacer algún cambio, ajuste o complementación al programa para que siga trabajando de manera correcta. Para poder realizar este trabajo se requiere que el programa este correctamente documentado.

CAPITULO II.

ENTIDADES PRIMITIVAS PARA EL DESARROLLO DE ALGORITMOS

2.1 Tipos de datos

2.2 Expresiones

2.3 Operadores y operandos

2.4 Identificadores como localidades de memoria

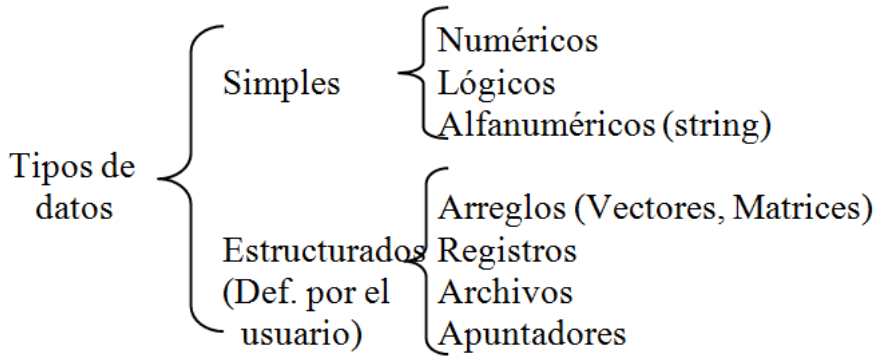
OBJETIVO EDUCACIONAL:

El alumno:

- Conocerá las reglas para cambiar fórmulas matemáticas a expresiones válidas para la computadora, además de diferenciar constantes e identificadores y tipos de datos simples.

2.1 Tipos De Datos

Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un simple carácter, tal como 'b', un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.



Tipos de Datos Simples

- ***Datos Numéricos:*** Permiten representar valores escalares de forma numérica, esto incluye a los números enteros y los reales. Este tipo de datos permiten realizar operaciones aritméticas comunes.

- ***Datos Lógicos:*** Son aquellos que solo pueden tener dos valores (cierto o falso) ya que representan el resultado de una comparación entre otros datos (numéricos o alfanuméricos).

- ***Datos Alfanuméricos (String):*** Es una secuencia de caracteres alfanuméricos que permiten representar valores identificables de forma descriptiva, esto incluye nombres de personas, direcciones, etc. Es posible representar números como alfanuméricos, pero estos pierden su propiedad matemática, es decir no es posible hacer operaciones con ellos. Este tipo de datos se representan encerrados entre comillas.

Ejemplo:

- “Servicio Nacional de Aprendizaje SENA”
- “2005”
- 2013
- False(0)
- True(1)

2.2 Expresiones

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales. Por ejemplo:

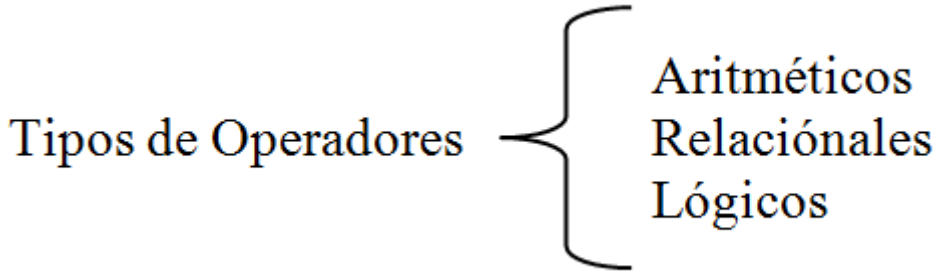
a+(b + 3)/c

Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones indicadas.
Una expresión consta de operadores y operandos. Según sea el tipo de datos que manipulan, se clasifican las expresiones en:

- Aritméticas
- Relaciónales
- Lógicas

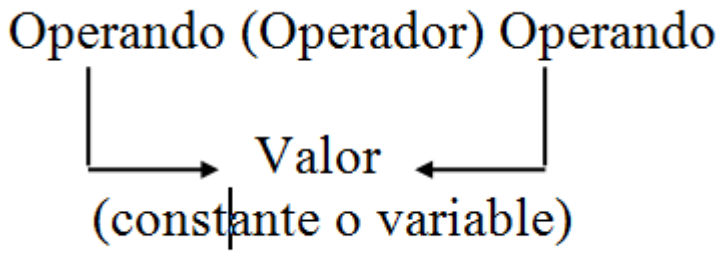
2.3 Operadores y Operandos

➢ ***Operadores:*** Son elementos que relacionan de forma diferente, los valores de una o más variables y/o constantes. Es decir, los operadores nos permiten manipular valores.



➢ ***Operadores Aritméticos:*** Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores (variables y constantes).

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real.



Operadores Aritméticos

+	Suma	
-	Resta	
*	Multiplicación	
/	División	
Mod	Modulo	(residuo de la división entera)

Ejemplos:

Expresión	Resultado
7 / 2	= 3.5
12 mod 7	= 5
4 + 2 * 5	= 14

Prioridad de los Operadores Aritméticos

- Todas las expresiones entre paréntesis se evalúan primero. Las expresiones con paréntesis anidados se evalúan de dentro a fuera, el paréntesis más interno se evalúa primero.
- Dentro de una misma expresión los operadores se evalúan en el siguiente orden.

  1.- ^ Exponenciación

  2.- *, /, mod Multiplicación, división, módulo.

  3.- +, - Suma y resta.

- Los operadores en una misma expresión con igual nivel de prioridad se evalúan de izquierda a derecha.

Ejemplos:

4 + 2 * 5 = 14	
23 * 2 / 5 = 9.2	46 / 5 = 9.2
3 + 5 * (10 - (2 + 4)) = 23	3 + 5 * (10 - 6) = 3 + 5 * 4 = 3 + 20 = 23
3.5 + 5.09 - 14.0 / 40 = 5.09	3.5 + 5.09 - 3.5 = 8.59 - 3.5 = 5.09

2.1 * (1.5 + 3.0 * 4.1) = 28.98 2.1 * (1.5 + 12.3) = 2.1 * 13.8 = 28.98

Operadores Relaciónales:

- Se utilizan para establecer una relación entre dos valores.
- Compara estos valores entre si y esta comparación produce un resultado de certeza o falsedad (verdadero o falso).
- Los operadores relaciónales comparan valores del mismo tipo (numéricos o cadenas)
- Tienen el mismo nivel de prioridad en su evaluación.
- Los operadores relaciónales tiene menor prioridad que los aritméticos.

Operadores Relaciónales

>	Mayor que
<	Menor que
> =	Mayor o igual que
< =	Menor o igual que
< >	Diferente
=	Igual

Ejemplos:

Si a = 10 b = 20 c = 30

- a + b > c Falso
- a - b < c Verdadero
- a - b = c Falso
- a * b < > c Verdadero

Ejemplos no lógicos:

- a < b < c
- 10 < 20 < 30
- T < 30 (no es lógico porque tiene diferentes operandos)

Operadores Lógicos:

η Estos operadores se utilizan para establecer relaciones entre valores lógicos.

η Estos valores pueden ser resultado de una expresión relacional.

Operadores Lógicos

And Y

Or O

Not Negación

Operador And

Operando1	*Operador*	*Operando2*	*Resultado*
T	AND	T	T

T	F	F
F	T	F
F	F	F

Operador Or

Operando1	*Operador*	*Operando2*	*Resultado*
T	OR	T	T
T		F	T
F		T	T
F		F	F

Operador Not

Operando	*Resultado*
T	F
F	T

Ejemplos:

(a < b) and (b < c)

(10<20) and (20<30)

T and T

| |

→ T ←

Prioridad de los Operadores Lógicos

Not

And

Or

Prioridad de los Operadores en General

1.- ( )

2.- ^

3.- *, /, Mod, Not

4.- +, -, And

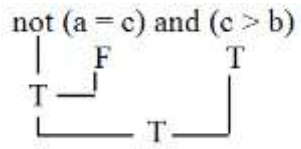
5.- >, <, > =, < =, < >, =, Or

Ejemplos:

a = 10 b = 12 c = 13 d =10

```
((a > b)or(a < c)) and ((a = c) or (a > = b))
   F        T          F        F
   └─ T ─┘             └─ F ─┘
      └──────  F  ──────────┘
```

```
((a > = b) or (a < d)) and (( a > = d) and (c > d))
    F          F              T           T
    └─  F  ─┘                 └──  T ─┘
        └──────────  F  ──────────────┘
```



2.4 Identificadores

Los identificadores representan los datos de un programa (constantes, variables, tipos de datos). Un identificador es una secuencia de caracteres que sirve para identificar una posición en la memoria de la computadora, que nos permite acceder a su contenido.

Ejemplo: Nombre

Num_hrs

Calif2

Reglas para formar un Identificador

- Debe comenzar con una letra (A - Z, mayúsculas o minúsculas).
- No deben contener espacios en blanco.
- No deben contener caracteres especiales (*,"#$%&/()=…).

Constantes y Variables

- **Constante**: Una constante es un dato numérico o alfanumérico que no cambia durante la ejecución del programa.

  Ejemplo:

  pi = 3.1416

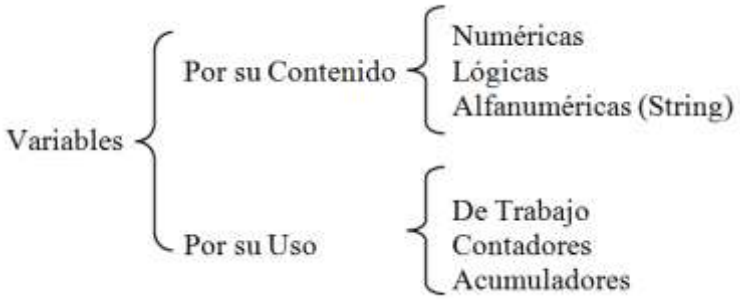
- **Variable**: Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambia durante la ejecución del programa. Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cuál podamos identificarla dentro de un algoritmo.

  Ejemplo:

  area = pi * radio ^ 2

  Las variables son: el radio, el área y la constate es pi

Clasificación de las Variables



Por su Contenido

- **Variable Numéricas**: Son aquellas en las cuales se almacenan valores numéricos, positivos o negativos, es decir almacenan números del 0 al 9, signos (+ y -) y el punto decimal. Ejemplo:

  iva=0.15 pi=3.1416 costo=2500

- **Variables Lógicas**: Son aquellas que solo pueden tener dos valores (cierto o falso) estos representan el resultado de una comparación entre otros datos.
- **Variables Alfanuméricas**: Esta formada por caracteres alfanuméricos (letras, números y caracteres especiales). Ejemplo:

  letra='a' apellido='lopez' direccion='Av. Libertad #190'

Por su Uso

- **Variables de Trabajo**: Variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa. Ejemplo:

  suma=a+b/c

- **Contadores**: Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Con los incrementos generalmente de uno en uno.
- **Acumuladores**: Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente.

CAPITULO III.

TÉCNICAS DE DISEÑO

3.1 Top down

3.2 Bottom up

OBJETIVO EDUCACIONAL:

El alumno:

- Conocerá las características de las técnicas de diseño más empleadas, así como su aplicación a cada tipo de problemas

3.1 Top Down

También conocida como de arriba-abajo y consiste en establecer una serie de niveles de mayor a menor complejidad (arriba-abajo) que den solución al problema. Consiste en efectuar una relación entre las etapas de la estructuración de forma que una etapa jerárquica y su inmediato inferior se relacionen mediante entradas y salidas de información.

Este diseño consiste en una serie de descomposiciones sucesivas del problema inicial, que recibe el refinamiento progresivo del repertorio de instrucciones que van a formar parte del programa.

La utilización de la técnica de diseño Top-Down tiene los siguientes objetivos básicos:

- Simplificación del problema y de los subprogramas de cada descomposición.

- Las diferentes partes del problema pueden ser programadas de modo independiente e incluso por diferentes personas.

- El programa final queda estructurado en forma de bloque o módulos lo que hace más sencilla su lectura y mantenimiento.

3.2 Bottom Up

El diseño ascendente se refiere a la identificación de aquellos procesos que necesitan computarizarse con forme vayan apareciendo, su análisis como sistema y su codificación, o bien, la adquisición de paquetes de software para satisfacer el problema inmediato.

Cuando la programación se realiza internamente y haciendo un enfoque ascendente, es difícil llegar a integrar los subsistemas al grado tal de que el desempeño global, sea fluido. Los problemas de integración entre los subsistemas son sumamente costosos y muchos de ellos no se solucionan hasta que la programación alcanza la fecha límite para la integración total del sistema. En esta fecha, ya se cuenta con muy poco tiempo, presupuesto o paciencia de los usuarios, como para corregir aquellas delicadas interfaces, que en un principio, se ignoran.

Aunque cada subsistema parece ofrecer lo que se requiere, cuando se contempla al sistema como una entidad global, adolece de ciertas limitaciones por haber tomado un enfoque ascendente. Uno de ellos es la duplicación de esfuerzos para accesar el software y más aun al introducir los datos. Otro es, que se introducen al sistema muchos datos carentes de valor. Un tercero y tal vez el más serio inconveniente del enfoque ascendente, es que los objetivos globales de la organización no fueron considerados y en consecuencia no se satisfacen.

CAPITULO IV.

TÉCNICAS PARA LA FÓRMULACIÓN DE ALGORITMOS

4.1 Diagrama de flujo

4.2 Pseudocódigo

4.3 Diagrama estructurado (nassi-schneiderman)

OBJETIVO EDUCACIONAL:

El alumno:

- Será capaz de diferenciar los métodos de representación y fórmulación de algoritmos, así como de conocer las características más importantes de cada técnica.

Las dos herramientas utilizadas comúnmente para diseñar algoritmos son:

Diagrama de Flujo

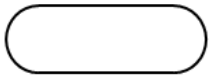
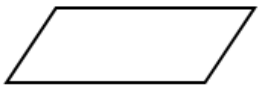
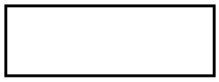
Pseudocódigo

4.1 Diagrama de Flujo

Un diagrama de flujo es la representación gráfica de un algoritmo. También se puede decir que es la representación detallada en forma gráfica de cómo deben realizarse los pasos en la computadora para producir resultados.

Esta representación gráfica se da cuando varios símbolos (que indican diferentes procesos en la computadora), se relacionan entre si mediante líneas que indican el orden en que se deben ejecutar los procesos.

Los símbolos utilizados han sido normalizados por el instituto norteamericano de normalización (ANSI).

SÍMBOLO	DESCRIPCIÓN
	Indica el inicio y el final de nuestro diagrama de flujo.
	Indica la entrada y salida de datos.
	Símbolo de proceso y nos indica la asignación de un valor en la memoria y/o la ejecución

	de una operación aritmética.
	Símbolo de decisión indica la realización de una comparación de valores.
	Se utiliza para representar los subprogramas.
	Conector dentro de página. Representa la continuidad del diagrama dentro de la misma página.
	Conector fuera de página. Representa la continuidad del diagrama en otra página.
	Indica la salida de información por impresora.
	Indica la salida de información en la pantalla o monitor.
	Líneas de flujo o dirección. Indican la Secuencia en que se realizan las operaciones.

Recomendaciones para el diseño de Diagramas de Flujo

- Se deben de usar solamente líneas de flujo horizontales y/o verticales.
- Se debe evitar el cruce de líneas utilizando los conectores.
- Se deben usar conectores solo cuando sea necesario.
- No deben quedar líneas de flujo son conectar.
- Se deben trazar los símbolos de manera que se puedan leer de arriba hacia abajo y de izquierda a derecha.

- Todo texto escrito dentro de un símbolo deberá ser escrito claramente, evitando el uso de muchas palabras.

4.2 Pseudocódigo

Mezcla de lenguaje de programación y español (o inglés o cualquier otro idioma) que se emplea, dentro de la programación estructurada, para realizar el diseño de un programa. En esencial, el pseudocódigo se puede definir como un lenguaje de especificaciones de algoritmos.

Es la representación narrativa de los pasos que debe seguir un algoritmo para dar solución a un problema determinado. El pseudocódigo utiliza palabras que indican el proceso a realizar.

Ventajas de utilizar un Pseudocódigo a un Diagrama de Flujo

- Ocupa menos espacio en una hoja de papel
- Permite representar en forma fácil operaciones repetitivas complejas
- Es muy fácil pasar de pseudocódigo a un programa en algún lenguaje de programación.
- Si se siguen las reglas se puede observar claramente los niveles que tiene cada operación.

4.3 Diagramas estructurados (Nassi-Schneiderman)

El diagrama estructurado N-S también conocido como diagrama de chapin es como un diagrama de flujo en el que se omiten las flechas de unión y las cajas son contiguas. Las acciones sucesivas se pueden escribir en cajas sucesivas y como en los diagramas de flujo, se pueden escribir diferentes acciones en una caja. Un algoritmo se represente en la sig. forma:

Inicio
Accion1
Accion2
. . .
Fin

Inicio
Leer Nombre,Hrs,Precio
Calcular Salario = Hrs * Precio
Calcular Imp = Salario* 0.15
Calcular Neto = Salario + Imp
Escribir Nombre, Imp, SNeto
Fin

CAPITULO V.

ESTRUCTURAS ALGORITMICAS

5.1 Secuenciales

- Asignación
- Entrada
- Salida

5.2 Condicionales

- Simples
- Múltiples

5.3 Repetición fila condicional

OBJETIVO EDUCACIONAL:

El alumno:

- Conocerá las diferentes estructuras algorítmicas como componentes básicos de los programas y aplicara la combinación de ellas para el desarrollo de algoritmos más complejos.

ESTRUCTURAS ALGORITMICAS

Las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas. Estas estructuras se clasifican de acuerdo con su complejidad en:

5.1. Estructuras Secuenciales

La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso. Una estructura secuencial se representa de la siguiente forma:

```
Inicio
 Accion1
 Accion2
  .
  .
 AccionN
Fin
```

- Asignación: La asignación consiste, en el paso de valores o resultados a una zona de la memoria. Dicha zona será reconocida con el nombre de la variable que recibe el valor. La asignación se puede clasificar de la siguiente forma:

- Simples: Consiste en pasar un valor constate a una variable (a=15)

- Contador: Consiste en usarla como un verificador del número de veces que se realiza un proceso (a=a+1)
- Acumulador: Consiste en usarla como un sumador en un proceso (a=a+b)
- De trabajo: Donde puede recibir el resultado de una operación matemática que involucre muchas variables (a=c+b*2/4).

- Lectura: La lectura consiste en recibir desde un dispositivo de entrada (p.ej. el teclado) un valor. Esta operación se representa en un pseudocódigo como sigue:

Leer a, b

Donde "a" y "b" son las variables que recibirán los valores

Escritura: Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje. Este proceso se representa en un pseudocódigo como sigue:

Escribe "Mi nombre es: ", nombre

Donde "El resultado es:" es un mensaje que se desea aparezca y R es una variable que contiene un valor.

5.1.1 Problemas Secuenciales

1) Suponga que un individuo desea invertir su capital en un banco y desea saber cuánto dinero ganará después de un mes si el banco paga a razón de 2% mensual.

```
Inicio
  Leer cap_inv
  gan = cap_inv * 0.02
  Imprimir gan
```

Fin

2) Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el vendedor desea saber cuánto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

```
Inicio
  Leer sb, v1, v2, v3
  tot_vta = v1 + v2 + v3
  com = tot_vta * 0.10
  tpag = sb + com
  Imprimir tpag, com
Fin
```

3) Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.

```
Inicio
  Leer tc
  d = tc * 0.15
  tp = tc - d
  Imprimir tp
Fin
```

4) Un alumno desea saber cuál será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:

55% del promedio de sus tres calificaciones parciales.

30% de la calificación del examen final.

15% de la calificación de un trabajo final.

ANÁLISIS

- Un alumno desea saber cuál será su **calificación final** en la materia de Algoritmos.
- Porcentajes:

✓ 55% del promedio de sus tres calificaciones parciales.
✓ 30% de la calificación del examen final.
✓ 15% de la calificación de un trabajo final.

Entrada

calificacion1, calificacion2, calificacion3, examenFinal, trabajoFinal: entera

Salida

promedio, porcentajeTresParciales, porcentajeExamenFinal, porcentajeTrabajoFinal, calificiacionFinal: entero

```
Inicio

  Leer c1, c2, c3, ef, tf

  prom = (c1 + c2 + c3)/3

  ppar = prom * 0.55

  pef = ef * 0.30

  ptf = tf * 0.15

  cf = ppar + pef + ptf

  Imprimir cf

Fin
```

5) Un maestro desea saber qué porcentaje de hombres y que porcentaje de mujeres hay en un grupo de estudiantes.

```
Inicio
  Leer nh, nm
  ta = nh + nm
  ph = nh * 100 / ta
  pm = nm * 100 / ta
  Imprimir ph, pm
Fin
```

6) Realizar un algoritmo que calcule la edad de una persona.

```
Inicio
  Leer fnac, fact
  edad = fact - fnac
  Imprimir edad
Fin.
```

5.1.2 Problemas Propuestos Secuenciales

1) Dada una cantidad en pesos, obtener la equivalencia en dólares, asumiendo que la unidad cambiaria es un dato desconocido.

2) Leer un número y escribir el valor absoluto del mismo.

3) La presión, el volumen y la temperatura de una masa de aire se relacionan por la fórmula:

masa = (presión * volumen)/(0.37 * (temperatura + 460))

4) Calcular el número de pulsaciones que una persona debe tener por cada 10 segundos de ejercicio, si la fórmula es:

num. pulsaciones = (220 - edad)/10

5) Calcular el nuevo salario de un obrero si obtuvo un incremento del 25% sobre su salario anterior.

6) En un hospital existen tres áreas: Ginecología, Pediatría, Traumatología. El presupuesto anual del hospital se reparte conforme a la sig. tabla:

Área	Porcentaje del presupuesto
Ginecología	40%
Traumatología	30%
Pediatría	30%

Obtener la cantidad de dinero que recibirá cada área, para cualquier monto presupuestal.

7) El dueño de una tienda compra un artículo a un precio determinado. Obtener el precio en que lo debe vender para obtener una ganancia del 30%.

8) Todos los lunes, miércoles y viernes, una persona corre la misma ruta y cronometra los tiempos obtenidos. Determinar el tiempo promedio que la persona tarda en recorrer la ruta en una semana cualquiera.

9) Tres personas deciden invertir su dinero para fundar una empresa. Cada una de ellas invierte una cantidad distinta. Obtener el porcentaje que cada quien invierte con respecto a la cantidad total invertida.

10) Un alumno desea saber cuál será su promedio general en las tres materias más difíciles que cursa y cuál será el promedio que obtendrá en cada una de ellas. Estas materias se evalúan como se muestra a continuación:

La calificación de Matemáticas se obtiene de la sig. manera:

Examen 90%

Promedio de tareas 10%

En esta materia se pidió un total de tres tareas.

La calificación de Física se obtiene de la sig. manera:

Examen 80%

Promedio de tareas 20%

En esta materia se pidió un total de dos tareas.

La calificación de Química se obtiene de la sig. manera:

Examen 85%

Promedio de tareas 15%

En esta materia se pidió un promedio de tres tareas.

5.2 Estructuras de Condicionales

Las estructuras condicionales comparan una variable contra otro(s) valor(es), para que en base al resultado de esta comparación, se siga un curso de acción dentro del programa. Cabe mencionar que la comparación se puede hacer contra otra variable o contra una constante, según se necesite. Existen dos tipos básicos, las simples y las múltiples.

- **Simples**: Las estructuras condicionales simples se les conoce como "Tomas de decisión". Estas tomas de decisión tienen la siguiente forma:

Si <condición> entonces

    Acción(es)

Fin-si

- **Dobles**: Las estructuras condicionales dobles permiten elegir entre dos opciones o alternativas posibles en función del cumplimiento o no de una determinada condición. Se representa de la siguiente forma:

Si <condición> entonces

Acción(es)

si no

Acción(es)

Fin-si

Dónde:

Si …… Indica el comando de comparación

Condición…… Indica la condición a evaluar

entonces…….…… Precede a las acciones a realizar cuando se cumple la condición

acción(es)…… Son las acciones a realizar cuando se cumple o no la condición

si no…… Precede a las acciones a realizar cuando no se cumple la condición

Dependiendo de si la comparación es cierta o falsa, se pueden realizar una o más acciones.

- **Múltiples**: Las estructuras de comparación múltiples, son tomas de decisión especializadas que permiten comparar una variable contra distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas. La forma común es la siguiente:

```
Si <condición> entonces
        Acción(es)
    si no
        Si <condición> entonces
                Acción(es)
            si no
                .
                .
                .
```

Varias condiciones

- Forma General

  Casos Variable

  Op1: Acción(es)

  Op2: Acción(es)

  .

  .

  OpN: acción

  Fin-casos

5.2.1 Problemas Condicionales

5.2.1.1 Problemas Selectivos Simples

1) Un hombre desea saber cuánto dinero se genera por concepto de intereses sobre la cantidad que tiene en inversión en el banco. El decidirá reinvertir los intereses siempre y cuando estos excedan a $7000, y en ese caso desea saber cuánto dinero tendrá finalmente en su cuenta.
   - Cuánto dinero se genera por concepto de intereses sobre la cantidad que tiene en inversión en el banco.
   - El decidirá reinvertir los intereses siempre y cuando estos excedan a $7000, y en ese caso desea saber cuánto dinero tendrá finalmente en su cuenta.

Entrada

cantidadInvertida: Double

tasaInteres: Double

Salidad

interes: Double

dineroFinal: Double

```
Entero cap, p_int

Real int, capf=0

Inicio

  Leer p_int, cap

  int = cap * p_int

  si int > 7000 entonces

        capf = cap + int

   fin-si

   Imprimir capf

fin
```

2) Determinar si un alumno aprueba a reprueba un curso, sabiendo que aprobara si su promedio de tres calificaciones es mayor o igual a 70; reprueba en caso contrario.

```
Entero calif1, calif2, calif3
```

```
Real prom

Inicio

  Leer calif1, calif2, calif3

  prom = (calif1 + calif2 + calif3)/3

  Si prom >= 70 entonces

        Imprimir "alumno aprobado"

  si no

        Imprimir "alumno reprobado"

  Fin-si

Fin
```

3) En un almacén se hace un 20% de descuento a los clientes cuya compra supere los $1000 ¿ Cuál será la cantidad que pagara una persona por su compra?

```
Entero compra

Real desc, tot_pag

Inicio
        Leer compra
        Si compra > 1000 entonces
                desc = compra * 0.20
        si no
                desc = 0
        fin-si
        tot_pag = compra - desc
        Imprimir tot_pag
fin
```

4) Un obrero necesita calcular su salario semanal, el cual se obtiene de la sig. manera:

Si trabaja 40 horas o menos se le paga $16 por hora

Si trabaja más de 40 horas se le paga $16 por cada una de las primeras 40 horas y $20 por cada hora extra.

```
Inicio
  Leer ht
  Si ht > 40 entonces
        he = ht - 40
        ss = he * 20 + 40 * 16
     si no
        ss = ht * 16
  Fin-si
  Imprimir ss
Fin
```

5) Un hombre desea saber cuánto dinero se genera por concepto de intereses sobre la cantidad que tiene en inversión en el banco. El decidirá reinvertir los intereses siempre y cuando estos excedan a $7000, y en ese caso desea saber cuánto dinero tendrá finalmente en su cuenta.

```
Inicio
  Leer p_int, cap
  int = cap * p_int
  si int > 7000 entonces
        capf = cap + int
   fin-si
   Imprimir capf
fin
```

6) Que lea dos números y los imprima en forma ascendente

```
Inicio
```

```
  Leer num1, num2
  Si num1 < num2 entonces
        Imprimir num1, num2
    si no
        Imprimir num2, num1
  fin-si
fin
```

7) Una persona enferma, que pesa 70 kg, se encuentra en reposo y desea saber cuántas calorías consume su cuerpo durante todo el tiempo que realice una misma actividad. Las actividades que tiene permitido realizar son únicamente dormir o estar sentado en reposo. Los datos que tiene son que estando dormido consume 1.08 calorías por minuto y estando sentado en reposo consume 1.66 calorías por minuto.

```
Inicio
  Leer act$, tiemp
  Si act$ = "dormido" entonces
        cg = 1.08 * tiemp
    si no
        cg = 1.66 * tiemp
  fin-si
  Imprimir cg
Fin
```

8) Hacer un algoritmo que imprima el nombre de un artículo, clave, precio original y su precio con descuento. El descuento lo hace en base a la clave, si la clave es 01 el descuento es del 10% y si la clave es 02 el descuento en del 20% (solo existen dos claves).

```
Inicio
  Leer nomb, cve, prec_orig
  Si cve = 01 entonces
```

```
            prec_desc = prec_orig - prec_orig * 0.10


        si no

            prec_desc = prec_orig - prec_orig * 0.20

    fin-si

    Imprimir nomb, cve, prec_orig, prec_desc

  fin
```

9) Hacer un algoritmo que calcule el total a pagar por la compra de camisas. Si se compran tres camisas o más se aplica un descuento del 20% sobre el total de la compra y si son menos de tres camisas un descuento del 10%

```
  Inicio

    Leer num_camisas, prec

    tot_comp = num_camisas * prec

    Si num_camisas > = 3 entonces

            tot_pag = tot_comp - tot_comp * 0.20

        si no

            tot_pag = tot_comp - tot_comp * 0.10

    fin-si

    Imprimir tot_pag

  fin
```

10) Una empresa quiere hacer una compra de varias piezas de la misma clase a una fabrica de refacciones. La empresa, dependiendo del monto total de la compra, decidirá que hacer para pagar al fabricante.

Si el monto total de la compra excede de $500 000 la empresa tendrá la capacidad de invertir de su propio dinero un 55% del monto de la compra, pedir prestado al banco un 30% y el resto lo pagara solicitando un crédito al fabricante.

Si el monto total de la compra no excede de $500 000 la empresa tendrá capacidad de invertir de su propio dinero un 70% y el restante 30% lo pagara solicitando crédito al fabricante.

El fabricante cobra por concepto de intereses un 20% sobre la cantidad que se le pague a crédito.

```
Inicio
  Leer costopza, numpza
  totcomp = costopza * numpza
  Si totcomp > 500 000 entonces
        cantinv = totcomp * 0.55
        préstamo = totcomp * 0.30
        crédito = totcomp * 0.15
    si no
        cantinv = totcomp * 0.70
        crédito = totcomp * 0.30
        préstamo = 0
  fin-si
  int = crédito * 0.20
  Imprimir cantinv, préstamo, crédito, int
Fin
```

5.2.1.2 Problemas Propuesto

1) Calcular el total que una persona debe pagar en un llantera, si el precio de cada llanta es de $800 si se compran menos de 5 llantas y de $700 si se compran 5 o más.

2) En un supermercado se hace una promoción, mediante la cual el cliente obtiene un descuento dependiendo de un número que se escoge al azar. Si el número escogido es menor que 74 el descuento es del 15% sobre el total de la compra, si es mayor o igual a 74 el descuento es del 20%. Obtener cuánto dinero se le descuenta.

3) Calcular el número de pulsaciones que debe tener una persona por cada 10 segundos de ejercicio aeróbico; la fórmula que se aplica cuando el sexo es femenino es:

num. pulsaciones = (220 - edad)/10

y si el sexo es masculino:

num. pulsaciones = (210 - edad)/10

4) Una compañía de seguros está abriendo un depto. de finanzas y estableció un programa para captar clientes, que consiste en lo siguiente: Si el monto por el que se efectúa la fianza es menor que $50 000 la cuota a pagar será por el 3% del monto, y si el monto es mayor que $50 000 la cuota a pagar será el 2% del monto. La afianzadora desea determinar cuál será la cuota que debe pagar un cliente.

5) En una escuela la colegiatura de los alumnos se determina según el número de materias que cursan. El costo de todas las materias es el mismo.

Se ha establecido un programa para estimular a los alumnos, el cual consiste en lo siguiente: si el promedio obtenido por un alumno en el último periodo es mayor o igual que 9, se le hará un descuento del 30% sobre la colegiatura y no se le cobrara IVA; si el promedio obtenido es menor que 9 deberá pagar la colegiatura completa, la cual incluye el 10% de IVA.

Obtener cuanto debe pagar un alumno.

6) Una empresa de bienes raíces ofrece casas de interés social, bajo las siguientes condiciones: Si los ingresos del comprador son menores de $8000 o más el enganche será del 15% del costo de la casa y el resto se distribuirá en pagos mensuales, a pagar en diez años. Si los ingresos del comprador son menos de $8000 o más el enganche será del 30% del costo de la casa y el resto se distribuirá en pagos mensuales a pagar en 7 años.

La empresa quiere obtener cuanto debe pagar un comprador por concepto de enganche y cuanto por cada pago parcial.

7) El gobierno ha establecido el programa SAR (Sistema de Ahorro para el Retiro) que consiste en que los dueños de la empresa deben obligatoriamente depositar en una cuenta bancaria un porcentaje del salario de los trabajadores; adicionalmente los trabajadores pueden solicitar a la empresa que deposite directamente una cuota fija o un porcentaje de su salario en la cuenta del SAR, la cual le será descontada de su pago.

Un trabajador que ha decidido aportar a su cuenta del SAR desea saber la cantidad total de dinero que estará depositado a esa cuenta cada mes, y el pago mensual que recibirá.

8) Una persona desea iniciar un negocio, para lo cual piensa verificar cuánto dinero le prestara el banco por hipotecar su casa. Tiene una cuenta bancaria, pero no quiere disponer de ella a menos que el monto por hipotecar su casa sea muy pequeño. Si el monto de la hipoteca es menor que $1 000 000 entonces invertirá el 50% de la inversión total y un socio invertirá el otro 50%. Si el monto de la hipoteca es de $ 1 000 000 o más, entonces invertirá el monto total de la hipoteca y el resto del dinero que se necesite para cubrir la inversión total se repartirá a partes iguales entre el socio y el.

9) El gobierno del estado de México desea reforestar un bosque que mide determinado número de hectáreas. Si la superficie del terreno excede a 1 millón de metros cuadrados, entonces decidirá sembrar de la sig. manera:

Porcentaje de la superficie del bosque	Tipo de árbol
70%	pino
20%	oyamel
10%	cedro

Si la superficie del terreno es menor o igual a un millón de metros cuadrados, entonces decidirá sembrar de la sig. manera:

Porcentaje de la superficie del bosque	Tipo de árbol
50%	pino
30%	oyamel

20% cedro

El gobierno desea saber el número de pinos, oyameles y cedros que tendrá que sembrar en el bosque, si se sabe que en 10 metros cuadrados caben 8 pinos, en 15 metros cuadrados caben 15 oyameles y en 18 metros cuadrados caben 10 cedros. También se sabe que una hectárea equivale a 10 mil metros cuadrados.

10) Una fábrica ha sido sometida a un programa de control de contaminación para lo cual se efectúa una revisión de los puntos IMECA generados por la fábrica. El programa de control de contaminación consiste en medir los puntos IMECA que emite la fábrica en cinco días de una semana y si el promedio es superior a los 170 puntos entonces tendrá la sanción de parar su producción por una semana y una multa del 50% de las ganancias diarias cuando no se detiene la producción. Si el promedio obtenido de puntos IMECA es de 170 o menor entonces no tendrá ni sanción ni multa. El dueño de la fábrica desea saber cuánto dinero perderá después de ser sometido a la revisión.

11) Una persona se encuentra con un problema de comprar un automóvil o un terreno, los cuales cuestan exactamente lo mismo. Sabe que mientras el automóvil se devalúa, con el terreno sucede lo contrario. Esta persona comprará el automóvil si al cabo de tres años la devaluación de este no es mayor que la mitad del incremento del valor del terreno. Ayúdale a esta persona a determinar si debe o no comprar el automóvil.

1. Requisitos

- El terreno y el vehículo cuestan exactamente lo mismo.

- El automóvil se devalúa

- El terreno incrementa su valor

- Pasados tres años, la devaluación del vehículo no es mayor (<=) a la mitad del incremento (incremento/2)

2. Análisis

- Solicitamos valorVehiculo, valorTerreno, porcentajeDevaluacionVehiculo, porcentajeIncrementoTerreno

- Obtenemos el valor de la devaluación del vehículo para el primer año (devaluacionPrimerAnio = 1000.000*10%)

- Obtenemos el valor de la devaluación del vehículo para el segundo año (devaluacionSegundoAnio = 900.000*10%)

- Obtenemos el valor de la devaluación del vehículo para el tercer año (devaluacionTercerAnio = 810.000*10%)

- Obtenemos el valor del incremento del terreno para el primer año (incrementoPrimerAnio = 1000.000*10%)

- Obtenemos el valor del incremento del terreno para el segundo año (incrementoSegundoAnio = 1100.000*10%)

- Obtenemos el valor del incremento del terreno para el tercer año (incrementoTercerAnio = 1.110.000*10%)

- Sumamos las tres devaluaciones

- Sumamos los tres incrementos

- Dividimos el valor total del incremento

- Si la devaluación es <= a la mitad del incremento compra vehículo de los contrario compra terreno.

Tipo	Variable	Descripción	Operación
Real	valorVehiculo	Almacena el valor del vehículo	Solicitar
Real	valorTerreno	Almacena el valor del terreno	Solicitar
Real	porcentajeDevaluacionVehiculo	Almacena el porcentaje de la devaluación del vehículo	Solicitar
Real	porcentajeIncrementoTerreno	Almacena el porcentaje de la incremento del terreno	Solicitar
Real	valorDevaluacionVehiculo	Almacena el valor de la devaluación del vehículo	valorVehiculo * porcentajeDevaluacionVehiculo
Real	valorIncrementoTerreno	Almacena el valor de la incremento	valorTerreno * porcentajeIncrementoTerreno

		del terreno	
Real	sumatoriaIncremento	Almacena el valor de los incrementos de los tres años	Incremento1 + incremento2 + incremento3
Real	sumatoriaDevaluacion	Almacena el valor de las devaluaciones de los tres años	devaluacion1+ Devaluacion2+ Devaluacion3

5.2.1.3 Problemas Selectivos Compuestos

1) Leer 2 números; si son iguales que los multiplique, si el primero es mayor que el segundo que los reste y si no que los sume.

```
Inicio
  Leer num1, num2
  si num1 = num2 entonces
        resul = num1 * num2
    si no
        si num1 > num2 entonces
              resul = num1 - num2
            si no
              resul = num1 + num2
        fin-si
  fin-si
fin
```

2) Leer tres números diferentes e imprimir el número mayor de los tres.

```
Inicio
  Leer num1, num2, num3
  Si (num1 > num2) and (num1 > num3) entonces
        mayor = num1
     si no
        Si (num2 > num1) and (num2 > num3) entonces
              mayor = num2
           si no
              mayor = num3
        fin-si
  fin-si
  Imprimir mayor
fin
```

3) Determinar la cantidad de dinero que recibirá un trabajador por concepto de las horas extras trabajadas en una empresa, sabiendo que cuando las horas de trabajo exceden de 40, el resto se consideran horas extras y que estas se pagan al doble de una hora normal cuando no exceden de 8; si las horas extras exceden de 8 se pagan las primeras 8 al doble de lo que se pagan las horas normales y el resto al triple.

```
Inicio
  Leer ht, pph
  Si ht < = 40 entonces
        tp = ht * pph
     si no
        he = ht - 40
        Si he < = 8 entonces
              pe = he * pph * 2
           si no
```

```
                pd = 8 * pph * 2
                pt = (he - 8) * pph * 3
                pe = pd + pt
            fin-si
            tp = 40 * pph + pe
        fin-si
        Imprimir tp
    fin
```

4) Calcular la utilidad que un trabajador recibe en el reparto anual de utilidades si este se le asigna como un porcentaje de su salario mensual que depende de su antigüedad en la empresa de acuerdo con la sig. tabla:

Tiempo	Utilidad
Menos de 1 año	5 % del salario
1 año o más y menos de 2 años	7% del salario
2 años o más y menos de 5 años	10% del salario
5 años o más y menos de 10 años	15% del salario
10 años o más	20% del salario

```
Inicio
  Leer sm, antig
  Si antig < 1 entonces
        util = sm * 0.05
      si no
        Si (antig > = 1) and (antig < 2) entonces
              util = sm * 0.07
            si no
```

```
                    Si (antig > = 2) and (antig < 5) entonces
                        util = sm * 0.10
                    si no
                        Si (antig > = 5) and (antig < 10) entonces
                            util = sm * 0.15
                        si no
                            util = sm * 0.20
                        fin-si
                    fin-si
            fin-si
    fin-si
                Imprimir util
fin
```

5) En una tienda de descuento se efectúa una promoción en la cual se hace un descuento sobre el valor de la compra total según el color de la bolita que el cliente saque al pagar en caja. Si la bolita es de color blanco no se le hará descuento alguno, si es verde se le hará un 10% de descuento, si es amarilla un 25%, si es azul un 50% y si es roja un 100%. Determinar la cantidad final que el cliente deberá pagar por su compra. se sabe que solo hay bolitas de los colores mencionados.

```
Inicio
    leer tc, b$
    si b$ = 'blanca' entonces
        d=0
    si no
    si b$ = 'verde' entonces
        d=tc*0.10
    si no
```

```
        si b$ = 'amarilla' entonces
            d=tc*0.25
          si no
          si b$ = 'azul' entonces
              d=tc*0.50
            si no
            d=tc
        fin-si
      fin-si
    fin-si
  fin-si
  fin
```

6) El IMSS requiere clasificar a las personas que se jubilaran en el año de 1997. Existen tres tipos de jubilaciones: por edad, por antigüedad joven y por antigüedad adulta. Las personas adscritas a la jubilación por edad deben tener 60 años o más y una antigüedad en su empleo de menos de 25 años. Las personas adscritas a la jubilación por antigüedad joven deben tener menos de 60 años y una antigüedad en su empleo de 25 años o más.

Las personas adscritas a la jubilación por antigüedad adulta deben tener 60 años o más y una antigüedad en su empleo de 25 años o más.

Determinar en qué tipo de jubilación, quedara adscrita una persona.

```
Inicio
  leer edad,ant
  si edad >= 60 and ant < 25 entonces
     imprimir "la jubilación es por edad"
    si no
       si edad >= 60 and ant > 25 entonces
```

```
            imprimir "la jubilación es por edad adulta"
        si no
            si edad < 60 and ant > 25 entonces
                imprimir "la jubilación es por antigüedad joven"
            si no
                imprimir "no tiene por qué jubilarse"
            fin-si
        fin-si
    fin-si
fin
```

5.2.2 Problemas Propuestos

1) En una fábrica de computadoras se planea ofrecer a los clientes un descuento que dependerá del número de computadoras que compre. Si las computadoras son menos de cinco se les dará un 10% de descuento sobre el total de la compra; si el número de computadoras es mayor o igual a cinco pero menos de diez se le otorga un 20% de descuento; y si son 10 o más se les da un 40% de descuento. El precio de cada computadora es de $11,000

2) En una llantera se ha establecido una promoción de las llantas marca "Ponchadas", dicha promoción consiste en lo siguiente:

Si se compran menos de cinco llantas el precio es de $300 cada una, de $250 si se compran de cinco a 10 y de $200 si se compran más de 10.

Obtener la cantidad de dinero que una persona tiene que pagar por cada una de las llantas que compra y la que tiene que pagar por el total de la compra.

3) En un juego de preguntas a las que se responde "Si" o "No" gana quien responda correctamente las tres preguntas. Si se responde mal a cualquiera de ellas ya no se pregunta la siguiente y termina el juego. Las preguntas son:

1. Colon descubrió América?

2. La independencia de México fue en el año 1810?

3. The Doors fue un grupo de rock Americano?

4) Un proveedor de estéreos ofrece un descuento del 10% sobre el precio sin IVA, de algún aparato si este cuesta $2000 o más. Además, independientemente de esto, ofrece un 5% de descuento si la marca es "NOSY". Determinar cuánto pagara, con IVA incluido, un cliente cualquiera por la compra de su aparato.

5) Una frutería ofrece las manzanas con descuento según la siguiente tabla:

NUM. DE KILOS COMPRADOS	% DESCUENTO
0 - 2	0%
2.01 - 5	10%
5.01 - 10	15%
10.01 en adelante	20%

Determinar cuánto pagara una persona que compre manzanas

es esa frutería.

Variable	Tipo	Descripción	Operación
numeroKilos	Decimal	Almacena el número de kilos comprados	Solicitado
valorKilo	Entero	Almacena el valor del kilo de manzanas	Constante
subTotal	Decimal	Almacena el valor a pagar antes del descuento	valorKilo* numeroKilos
porcentajeDescuento	Decimal	Almacena el porcentaje del descuento	If(numeroKilos<=2){ porcentajeDescuento = 0 }else if(numeroKilos>2 && numeroKilos<=5){ porcentajeDescuento = 0.1; }…
descuento	Decimal	Almacena el valor del descuento	subTotal * porcentajeDescuento

totalPagar	Decimal	Almacena el total a pagar	subTotal- descuento

6) El dueño de una empresa desea planificar las decisiones financieras que tomara en el siguiente año. La manera de planificarlas depende de lo siguiente:

Si actualmente su capital se encuentra con saldo negativo, pedirá un préstamo bancario para que su nuevo saldo sea de $10 000. Si su capital tiene actualmente un saldo positivo pedirá un préstamo bancario para tener un nuevo saldo de $20 000, pero si su capital tiene actualmente un saldo superior a los $20 000 no pedirá ningún préstamo.

Posteriormente repartirá su presupuesto de la siguiente manera.

$5 000 para equipo de computo

$2 000 para mobiliario

y el resto la mitad será para la compra de insumos y la otra para otorgar incentivos al personal.

Desplegar que cantidades se destinaran para la compra de insumos e incentivos al personal y, en caso de que fuera necesario, a cuánto ascendería la cantidad que se pediría al banco.

7) Tomando como base los resultados obtenidos en un laboratorio de análisis clínicos, un médico determina si una persona tiene anemia o no, lo cual depende de su nivel de hemoglobina en la sangre, de su edad y de su sexo. Si el nivel de hemoglobina que tiene una persona es menor que el rango que le corresponde, se determina su resultado como positivo y en caso contrario como negativo. La tabla en la que el medico se basa para obtener el resultado es la siguiente:

EDAD	NIVEL HEMOGLOBINA
0 - 1 mes	13 - 26 g%
> 1 y < = 6 meses	10 - 18 g%
> 6 y < = 12 meses	11 - 15 g%
> 1 y < = 5 años	11.5 - 15 g%
> 5 y < = 10 años	12.6 - 15.5 g%

> 10 y < = 15 años	13 - 15.5 g%
mujeres > 15 años	12 - 16 g%
hombres > 15 años	14 - 18 g%

8) Una institución educativa estableció un programa para estimular a los alumnos con buen rendimiento académico y que consiste en lo siguiente:

Si el promedio es de 9.5 o más y el alumno es de preparatoria, entonces este podrá cursar 55 unidades y se le hará un 25% de descuento.

Si el promedio es mayor o igual a 9 pero menor que 9.5 y el alumno es de preparatoria, entonces este podrá cursar 50 unidades y se le hará un 10% de descuento.

Si el promedio es mayor que 7 y menor que 9 y el alumno es de preparatoria, este podrá cursar 50 unidades y no tendrá ningún descuento.

Si el promedio es de 7 o menor, el número de materias reprobadas es de 0 a 3 y el alumno es de preparatoria, entonces podrá cursar 45 unidades y no tendrá descuento.

Si el promedio es de 7 o menor, el número de materias reprobadas es de 4 o más y el alumno es de preparatoria, entonces podrá cursar 40 unidades y no tendrá ningún descuento.

Si el promedio es mayor o igual a 9.5 y el alumno es de profesional, entonces podrá cursar 55 unidades y se le hará un 20% de descuento.

Si el promedio es menor de 9.5 y el alumno es de profesional, entonces podrá cursar 55 unidades y no tendrá descuento.

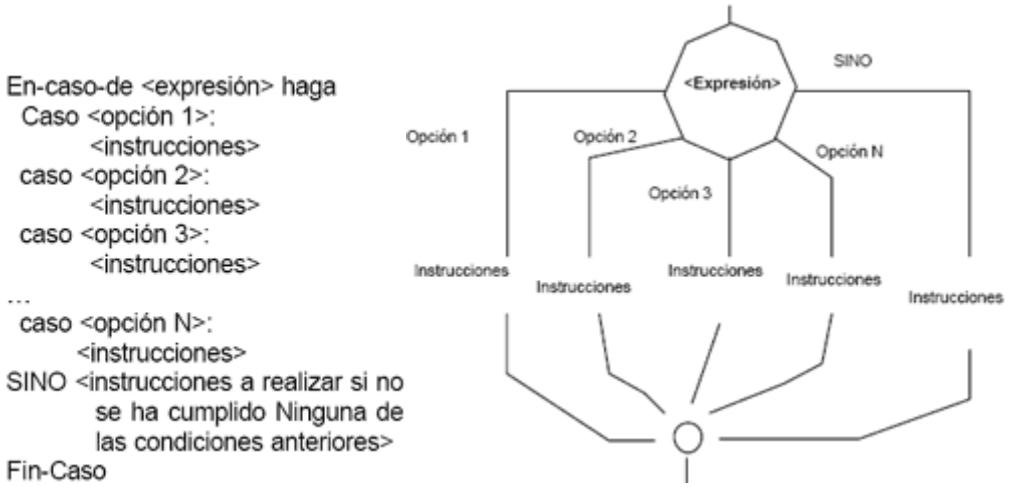
Obtener el total que tendrá que pagar un alumno si la colegiatura para alumnos de profesional es de $300 por cada cinco unidades y para alumnos de preparatoria es de $180 por cada cinco unidades.

9) Que lea tres números diferentes y determine el número medio del conjunto de los tres números (el número medio es aquel número que no es ni mayor, ni menor).

5.2.3 Condicionales Múltiples

Las estructuras de comparación múltiples, es una toma de decisión especializada que permiten evaluar una variable con distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas. La forma es la siguiente:

Pseudocódigo: Diagrama de flujo:



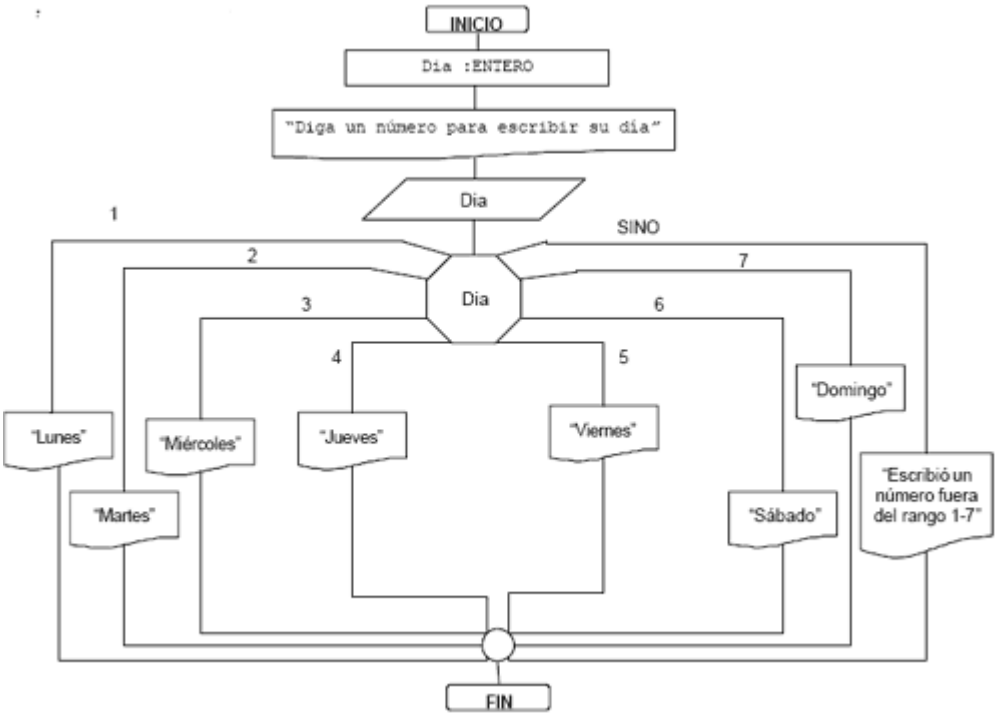
5.2.3.1 Problemas Condicionales Múltiples

Dado un numero entre 1 y 7 escriba su correspondiente día de la semana así:
1- Lunes 2- Martes 3- Miércoles 4- Jueves 5- Viernes 6- Sábado 7- Domingo
Exprese el algoritmo usando Pseudocódigo y diagrama de flujos.

Pseudocódigo: Pseudocódigo:

```
Inicio
  Entero Dia
  ESCRIBA "Diga un número para escribir su día"
  LEA Dia
  En caso de Dia haga
    Caso 1: ESCRIBA "Lunes"
    Caso 2: ESCRIBA "Martes"
    Caso 3: ESCRIBA "Miércoles"
    Caso 4: ESCRIBA "Jueves"
    Caso 5: ESCRIBA "Viernes"
    Caso 6: ESCRIBA "Sábado"
    Caso 7: ESCRIBA "Domingo"
   SINO: ESCRIBA "Escribió un numero fuera del rango 1-7"
  Fin caso
Fin
```

Diagrama de flujo:



5.2.3.2. Problemas propuestos

1) Leer un número de un dígito y escribir su correspondiente en letras.

2) Leer un número cualquiera y decir si es un dígito, en caso contrario no es de un dígito.

3) Leer un carácter cualquiera y decir si es una vocal, en caso contrario no es una vocal

4) Leer la temperatura de grados Celsius y convertirla a Kelvin o Fahrenheit.

5) Leer la temperatura de grados Kelvin y convertirla a Celsius o Fahrenheit.

6) Leer la temperatura de grados Fahrenheit y convertirla a Celsius, kelvin

5.3. Estructuras Cíclicas

Se llaman problemas repetitivos o cíclicos a aquellos en cuya solución es necesario utilizar un mismo conjunto de acciones que se puedan ejecutar una cantidad específica de veces. Esta cantidad puede ser fija (previamente determinada por el programador) o puede ser variable (estar en función de algún dato dentro del programa).Los ciclos se clasifican en:

5.3.1 Hacer-Para

Ciclos con un Número Determinado de Iteraciones.

Son aquellos en que el número de iteraciones se conoce antes de ejecutarse el ciclo. La forma de esta estructura es la siguiente:

```
Hacer para V.C = L.I a L.S
        Accion1
        Accion2
          .
          .
          .
        AccionN
Fin-para
```

Donde :

V.C	Variable de control del ciclo
L.I	Límite inferior
L.S	Límite superior

En este ciclo la variable de control toma el valor inicial del ciclo y el ciclo se repite hasta que la variable de control llegue al límite superior.

5.3.1.1 Problemas (Hacer para)

1) Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de Diseño Estructurado de Algoritmos

Inicio

Hacer para c = 1 a 5

Imprimir c

Fin para

Imprimir “Muchas gracias por utilizar el aplicativo”

Fin

Inicio

```
  Sum=0
  Leer Nom
  Hacer para c = 1 a 7
        Leer calif
        Sum = Sum + calif
  Fin-para
  prom = sum /7
  Imprimir prom
Fin.
```

2) Leer 10 números y obtener su cubo y su cuarta.

```
Inicio
  Hacer para n = 1 a 10
        Leer num
        cubo = num * num * num
        cuarta = cubo * num
        Imprimir cubo, cuarta
  Fin-para
Fin.
```

3) Leer 10 números e imprimir solamente los números positivos

```
Inicio
  Hacer para n = 1 a 10
```

```
        Leer num
        Si num > 0 entonces
            Imprimir num
        fin-si
    Fin-para
Fin.
```

4) Leer 20 números e imprimir cuantos son positivos, cuantos negativos y cuantos neutros.

```
Inicio
    cn = 0
    cp = 0
    cneg = 0
    Hacer para x = 1 a 20
        Leer num
        Sin num = 0 entonces
            cn = cn + 1
        si no
            Si num > 0 entonces
                cp = cp + 1
            si no
                cneg = cneg + 1
            Fin-si
        Fin-si
    Fin-para
    Imprimir cn, cp, cneg
Fin.
```

5) Leer 15 números negativos y convertirlos a positivos e imprimir dichos números.

```
Inicio
  Hacer para x = 1 a 15
        Leer num
        pos = num * -1
        Imprimir num, pos
  Fin-para
Fin.
```

6) Suponga que se tiene un conjunto de calificaciones de un grupo de 40 alumnos. Realizar un algoritmo para calcular la calificación media y la calificación más baja de todo el grupo.

```
Inicio
  sum = 0
  baja = 5
  Hacer para a = 1 a 40
        Leer calif
        sum = sum + calif
        Si calif < baja entonces
              baja = calif
        fin-si
  Fin-para
  media = sum / 2
  Imprimir media, baja
fin
```

7) Calcular e imprimir la tabla de multiplicar de un número cualquiera. Imprimir el multiplicando, el multiplicador y el producto.

```
Inicio
  Leer num                                                    60

  Hacer para X = 1 a 10
        resul = num * x
        Imprimir num, " * ", X, " = ", resul
  Fin-para
fin.
```

8) Simular el comportamiento de un reloj digital, imprimiendo la hora, minutos y segundos de un día desde las 0:00:00 horas hasta las 23:59:59 horas

Requerimientos

- Simular el comportamiento de un reloj digital
- imprimir
- la hora,
- minutos
- segundos de un día desde las 0:00:00 horas hasta las 23:59:59 horas

ANÁLISIS

- Necesitamos un ciclo que nos defina la hora de 0 a 23
- Necesitamos un ciclo que nos defina los minutos de 0 a 59
- Necesitamos un ciclo que nos defina los segundos de 0 a 59

Tipo	Operaciones	Variables	Descripción
entero	0a23	hora	hora
entero	0a59	minutos	minutos
entero	0a59	segundos	Segundos

```
Inicio
  Hacer para h = 1 a 23
    Hacer para m = 1 a 59
```

```
            Hacer para s = 1 a 59
              Imprimir h, m, s
            Fin-para
        Fin-para
      Fin-para
    fin.
```

5.3.1.2 Problemas Propuestos

1) Una persona debe realizar un muestreo con 50 personas para determinar el promedio de peso de los niños, jóvenes, adultos y viejos que existen en su zona habitacional. Se determinan las categorías con base en la sig. tabla:

CATEGORIA	EDAD
Niños	0 - 12
Jóvenes	13 - 29
Adultos	30 - 59
Viejos	60 en adelante

Requerimiento

- Realizar un muestreo con 50 personas
- determinar el promedio de peso
- de niños, jóvenes, adultos y viejos.

Análisis

- un ciclo de 1 a 50
- debo solicitar la edad del encuestado,
- debo solicitar el peso del encuestado.
- debemos evaluar si el encuestado es un niño, joven, adulto o adulto mayor.
- acumular en variables diferentes los pesos de niños, jóvenes, adultos y adultos mayores.

tipo	operación	variable	descripción
entero		control	Variable de control el ciclo hacer para
entero	solicitarla	edad	Edad de los encuestados
real	solicitarla	peso	Pesos de los encuestados
real	sumaNino=sumaNino+peso	sumaNino	Sumatoria o acumulados de los pesos de los niños
real	sumaJoven=sumaJoven+peso	sumaJoven	Sumatoria o acumulados de los pesos de los jóvenes
real	sumaAdult=sumaAdult+peso	sumaAdult	Sumatoria o acumulados de los pesos de los adultos
real	sumaAdultMayor= sumaAdultMayor+peso	sumaAdultMayor	Sumatoria o acumulados de los

			pesos de los adultos mayores
entero	cantiNino = cantiNino + 1	cantiNino	Cantidad niño
entero	cantiJoven = cantiJoven + 1	cantiJoven	Cantidad joven
entero	cantiAdult = cantiAdult + 1	cantiAdult	Cantidad adulto
entero	cantiAdultMayor = cantiAdultMayor + 1	cantiAdultMayor	Cantidad adulto mayor
real	promeNino=sumaNino/cantiNino	promeNino	Promedio niño
real	promeJoven=sumaJoven/cantiJoven	promeJoven	Promedio joven
real	promeAdult=sumaAdult/cantiAdult	promeAdult	Promedio adulto
real	promeAdultMayor= sumaAdultMayor/ cantiAdultMayor	promeAdultMayor	Promedio adulto mayor

2) Al cerrar un expendio de naranjas, 15 clientes que aún no han pagado recibirán un 15% de descuento si compran más de 10 kilos. Determinar cuánto pagara cada cliente y cuanto percibirá la tienda por esas compras.

3) En un centro de verificación de automóviles se desea saber el promedio de puntos contaminantes de los primeros 25 automóviles que lleguen. Así mismo se desea saber los puntos contaminantes del carro que menos contamino y del que más contamino.

4) Un entrenador le ha propuesto a un atleta recorrer una ruta de cinco kilómetros durante 10 días, para determinar si es apto para la prueba de 5 Kilómetros o debe buscar otra especialidad. Para considerarlo apto debe cumplir por lo menos una de las siguientes condiciones:

- Que en ninguna de las pruebas haga un tiempo mayor a 16 minutos.
- Que al menos en una de las pruebas realice un tiempo mayor a 16 minutos.
- Que su promedio de tiempos sea menor o igual a 15 minutos.

5) Un Zoólogo pretende determinar el porcentaje de animales que hay en las siguientes tres categorías de edades: de 0 a 1 año, de más de 1 año y menos de 3 y de 3 o más años. El zoológico todavía no está seguro del animal que va a estudiar. Si se decide por elefantes solo tomara una muestra de 20 de ellos; si se decide por las jirafas, tomara 15 muestras, y si son chimpancés tomara 40.

Ciclos con un Número Indeterminado de Iteraciones (Hacer-Mientras, Repetir-Hasta)

Son aquellos en que el número de iteraciones no se conoce con exactitud, ya que esta dado en función de un dato dentro del programa.

5.3.2 Hacer-Mientras

Esta es una estructura que repetirá un proceso durante "N" veces, donde "N" puede ser fijo o variable. Para esto, la instrucción se vale de una condición que es la que debe cumplirse para que se siga ejecutando. Cuando la condición ya no se cumple, entonces ya no se ejecuta el proceso. La forma de esta estructura es la siguiente:

```
Hacer mientras <condición>
        Accion1
        Accion2
          .
          .
        AccionN
Fin-mientras
```

Cadena respuesta;

Inicio

respuesta = "SI"

Hacer mientras respuesta = "SI"

Escriba "Desea iterar una vez más? (SI/NO)"

Lea respuesta;

Fin mientras

Escriba "Gracias por utilizar la aplicación."

Fin

respuesta	impresión
SI	Desea iterar una vez más? (SI/NO)
SI	Desea iterar una vez más? (SI/NO)
NO	Gracias por utilizar la aplicación.

Entero control

Inicio

control = 1

Hacer mientras control <= 5

Escriba control

control = control + 1

Fin mientras

Escriba “Gracias por utilizar la aplicación.”

Fin

control	Impresión
1	1
2	2
3	3
4	4
5	5
6	Gracias por utilizar la aplicación.

5.3.2.1 Problemas (Hacer Mientras)

1) Una compañía de seguros tiene contratados a n vendedores. Cada uno hace tres ventas a la semana. Su política de pagos es que un vendedor recibe un sueldo base, y un 10% extra por comisiones de sus ventas. El gerente de su compañía desea saber cuánto dinero obtendrá en la semana cada vendedor por concepto de comisiones por las tres ventas realizadas, y cuanto tomando en cuenta su sueldo base y sus comisiones.

REQUERIMIENTOS

- n vendedores
- Cada uno hace tres ventas a la semana
- un vendedor recibe un sueldo base
- un vendedor recibe un 10% extra por comisiones de sus ventas
- cuánto dinero obtendrá en la semana cada vendedor por concepto de comisiones por las tres ventas realizadas ?.
- y cuánto tomando en cuenta su sueldo base y sus comisiones .

ANÁLISIS

* Solicitar la cantidad de vendedores.

* Realizamos un ciclo para el número de vendedores.

* Solicitar el sueldo base(sb)

* Solicitar el valor de cada una de las ventas (v1, v2, v3).

* Obtener la comisión de cada una de las ventas (c1 = v1 * 10%, c2 = v2 * 10%, c3 = v3 * 10%)

* Obtener el total de comisiones(tcv = cv1+cv2+cv3)

* Obtener el total que recibo el vendedor (sb+tcv)

Tipo	Nombre	Descripción	Operación
Entero	cantidadVendedores		Solicitado
Entero	contador		Contador = 0; contador < cantidadVendedores; contador++
Decimal	sueldoBase		Solicitado
Decimal	venta1		Solicitado
Decimal	venta2		Solicitado
Decimal	venta3		Solicitado
Decimal	comisionVenta1		venta1 * 10%
Decimal	comisionVenta2		venta2 * 10%
Decimal	comisionVenta3		venta3 * 10%
Decimal	totalComision		comisionVenta1+ comisionVenta2+ comisionVenta3
Decimal	totalRecibVende		totalComision + sueldoBase

2) En una empresa se requiere calcular el salario semanal de cada uno de los n obreros que laboran en ella. El salario se obtiene de la sig. forma:

Si el obrero trabaja 40 horas o menos se le paga $20 por hora

Si trabaja más de 40 horas se le paga $20 por cada una de las primeras 40 horas y $25 por cada hora extra.

3) Determinar cuántos hombres y cuantas mujeres se encuentran en un grupo de n personas, suponiendo que los datos son extraídos alumno por alumno.

4) El Depto. de Seguridad Publica y Transito del D.F. desea saber, de los n autos que entran a la ciudad de México, cuantos entran con calcomanía de cada color. Conociendo el ultimo dígito de la placa de cada automóvil se puede determinar el color de la calcomanía utilizando la sig. relación:

DÍGITO	COLOR
1 o 2	amarilla
3 o 4	rosa
5 o 6	roja
7 o 8	verde
9 o 0	azul

5) Obtener el promedio de calificaciones de un grupo de n alumnos.

6) Una persona desea invertir su dinero en un banco, el cual le otorga un 2% de interés. Cuál será la cantidad de dinero que esta persona tendrá al cabo de un año si la ganancia de cada mes es reinvertida?

7) Calcular el promedio de edades de hombres, mujeres y de todo un grupo de alumnos.

8) Encontrar el menor valor de un conjunto de n números dados.

9) Encontrar el mayor valor de un conjunto de n números dados.

10) En un supermercado un cajero captura los precios de los artículos que los clientes compran e indica a cada cliente cuál es el monto de lo que deben pagar. Al final del día le indica a su supervisor cuanto fue lo que cobro en total a todos los clientes que pasaron por su caja.

11) Cinco miembros de un club contra la obesidad desean saber cuánto han bajado o subido de peso desde la última vez que se reunieron. Para esto se debe realizar un ritual de pesaje en donde cada uno se pesa en diez básculas distintas para así tener el promedio más exacto de su peso. Si existe diferencia positiva entre este promedio de peso y el peso de la última vez que se reunieron, significa que subieron de peso. Pero si la diferencia es negativa, significa que bajaron. Lo que el problema requiere es que por cada persona se imprima un letrero que diga: "SUBIO" o "BAJO" y la cantidad de kilos que subió o bajo de peso.

12) Se desea obtener el promedio de g grupos que están en un mismo año escolar; siendo que cada grupo puede tener n alumnos que cada alumno puede llevar m materias y que en todas las materias se promedian tres calificaciones para obtener el promedio de la materia. Lo que se desea desplegar es el promedio de los grupos, el promedio de cada grupo y el promedio de cada alumno.

5.3.3 Repetir-Hasta

Esta es una estructura similar en algunas características, a la anterior. Repite un proceso una cantidad de veces, pero a diferencia del Hacer-Mientras, el Repetir-Hasta lo hace hasta que la condición se cumple y no mientras, como en el Hacer-Mientras. Por otra parte, esta estructura permite realizar el proceso cuando menos una vez, ya que la condición se evalúa al final del proceso, mientras que en el Hacer-Mientras puede ser que nunca llegue a entrar si la condición no se cumple desde un principio. La forma de esta estructura es la siguiente:

```
Repetir
        Accion1
        Accion2
          .
          .
        AccionN
Hasta <condición>
```

5.3.3.1 Problemas Repetir – Hasta

1) En una tienda da descuento a las personas que van a pagar el importe de su compra llegan a la caja y sacan una bolita de color, que les dirá que descuento tendrán sobre el total de su compra. Determinar la cantidad que pagará cada cliente desde que la tienda abre hasta que cierra. Se sabe que si el color de la bolita es roja el cliente obtendrá un 40% de descuento; si es amarilla un 25% y si es blanca no obtendrá descuento.

1. Requerimientos

- Se ofrece un descuento del total de la compra que depende del color de una bola sacada por el cliente.

COLOR	% DESCUENTO
Roja	40
Amarilla	25
Blanca	0

- Obtener el total a pagar de cada cliente desde que la tienda abre hasta que cierra.

2. Análisis

- Realizar un ciclo repetir-hasta donde la condición dependerá de la respuesta del cliente
- Solicitar el valor total de la compra.
- Le solicitamos al cliente que saque la bolita para poder determinar el porcentaje de descuento.
- Obtener el porcentaje de descuento
- Obtenemos el valor a pagar restándole al valor total de la compra el valor del descuento.
- Mostrar el valor final a pagar.
- Preguntarle al cajero si ya se va a cerrar el almacén.
- Si la respuesta es positiva salimos de la aplicación.
- De lo contrario repetimos los pasos anteriores.

Tipo	Variable	Descripción	Operación
Real	totalCompr	Almacena el valor total de la compra de cada cliente	Solicitar
Cadena	colorBola	Almacena el color de la bola sacada por el cliente.	Solicitar
Real	porceDescu	Almacena el porcentaje del descuento a realizar al total de la compra	Si colorBola = “roja” entonces porceDescu = 0.4 si no…
Real	totalDescu	Almacena el descuento de cada compra.	totalCompr * porceDescu
Real	totalPagar	Almacena el valor que el cliente finalmente va a pagar.	totalCompr - totalDescu
Cadena	respuesta	Almacena la respuesta del cajero de sí la tiene continua abierta o no.	Solicitar

3. Diseño

4. Codificación

```
Real totalDescu, porceDescu, totalPagar
Cadena colorBola, respuesta
Inicio
        Repetir
                Imprimir "Señor usuario, por favor digite el  valor de su compra "
                Leer totalCompr
                Imprimir "Señor usuario, por favor digite el color de la bola sacada"
                Leer colorBola
                En caso de colorBola haga
                        Caso "roja" :
                                porceDescu = 0.4
                        romper
                        caso   "amarilla" :
                                porceDescu = 0.25
                        romper
                        caso "blanca" :
                                porceDescu = 0
                        sino
                                Imprimir "El color de la  bola digitada no existe por lo tanto el
                                porcentaje de descuento asignado es cero (0)"
                                porceDescu = 0
                Fin caso

                totalDescu = totalCompr * porceDescu

                totalPagar = totalCompr – totalDescu
                Imprimir "El total a pagar por la compra es : $ ",totalPagar
                Imprimir "Señor cajero el almacén continua abierto "
                Leer respuesta
        hasta (respuesta = "no")
Fin
```

5. Pruebas

2) un teatro otorga descuentos según la edad del cliente. Determinar la cantidad de dinero que el teatro deja de percibir por cada una de las categorías. Tomar en cuenta que los niños menores de 5 años no pueden entrar al teatro y que existe un precio único en los asientos. Los descuentos se hacen tomando en cuenta el siguiente cuadro:

	Edad	Descuento
Categoría 1	5 - 14	35 %

Categoría 2	15 - 19	25 %
Categoría 3	20 - 45	10 %
Categoría 4	46 - 65	25 %
Categoría 5	66 en adelante	35 %

3) Determinar la cantidad semanal de dinero que recibirá cada uno de los *n* obreros de una empresa. Se sabe que cuando las horas que trabajo un obrero exceden de 40, el resto se convierte en horas extras que se pagan al doble de una hora normal, cuando no exceden de 8; cuando las horas extras exceden de 8 se pagan las primeras 8 al doble de lo que se paga por una hora normal y el resto al triple.

4) En la Cámara de Diputados se levanta una encuesta con todos los integrantes con el fin de determinar qué porcentaje de los *n* diputados está a favor del Tratado de Libre Comercio, que porcentaje está en contra y que porcentaje se abstiene de opinar.

5) Una persona que va de compras a la tienda "Enano, S.A.", decide llevar un control sobre lo que va comprando, para saber la cantidad de dinero que tendrá que pagar al llegar a la caja. La tienda tiene una promoción del 20% de descuento sobre aquellos artículos cuya etiqueta sea roja. Determinar la cantidad de dinero que esta persona deberá pagar.

6) Un jefe de casilla desea determinar cuántas personas de cada una de las secciones que componen su zona asisten el día de las votaciones. Las secciones son: norte, sur y centro. También desea determinar cuál es la sección con mayor número de votantes.

7) Leer 50 calificaciones de un grupo de alumnos. Calcule y escriba el porcentaje de reprobados. Tomando en cuenta que la calificación mínima aprobatoria es de 70.

8) Suponga que tiene usted una tienda y desea registrar las ventas en su computadora. Diseñe un algoritmo que lea por cada cliente, el monto total de su compra. Al final del día que escriba la cantidad total de ventas y el número de clientes atendidos.

5.3.3.2 Problemas Repetitivos Compuestos

1) Suponga que tiene una tienda y desea registrar sus ventas por medio de una computadora. Diseñe un pseudocódigo que lea por cada cliente:

a) El monto de la venta.

b) Calcule e imprima el IVA.

c) Calcule e imprima el total a pagar.

d) Lea la cantidad con que paga el cliente,

e) Calcule e imprime el cambio.

Al final del día deberá imprimir la cantidad de dinero que debe haber en la caja.

2) Modificar el pseudocódigo anterior de tal forma que no permita que la cantidad con la que paga el cliente sea menor a lo que debe pagar.

3) Se tiene un conjunto de 1,000 tarjetas cada una contiene la información del censo para una persona:

- Número de censo,
- Sexo
- Edad
- Estado civil (a.- soltero, b. Casado, c. Viudo, d. Divorciado)

Diseñe un pseudocódigo estructurado que lea todos estos datos, e imprima el número de censo de todas las jóvenes solteras que estén entre 16 y 21 años.

4) Diseñe un pseudocódigo que calcule e imprima el pago de 102 trabajadores que laboran en la Cía. GACMAN. Los datos que se leerán serán los siguientes:

a) Las horas trabajadas

b) El sueldo por hora

c) El tipo de trabajador (1.-obrero,2.-empleado)

Para calcular los pagos considerar lo siguiente:

- Los obreros pagan 10 % de impuesto

- Los empleados pagan 10 % de impuesto.

- Los trabajadores (obreros y empleados) que reciban un pago menor de 100,000 pesos no pagan impuesto.

-Al final se deberá imprimir el total a pagar a los trabajadores y a los empleados.

5) Un grupo de 100 estudiantes presentan un examen de Física. Diseñe un diagrama que lea por cada estudiante la calificación obtenida y calcule e imprima:

- La cantidad de estudiantes que obtuvieron una calificación menor a 50.

- La cantidad de estudiantes que obtuvieron una calificación de 50 o más pero menor que 80.
- La cantidad de estudiantes que obtuvieron una calificación de 70 o más pero menor que 80.
- La cantidad de estudiantes que obtuvieron una calificación de 80 o más.

VI ARREGLOS

Un arreglo (array) es una colección de datos del mismo tipo, que se almacenan en posiciones consecutivas de memoria y reciben un nombre común. Para referirse a un determinado elemento de un array se deberá utilizar un índice, que especifique su posición relativa en el array. Un arreglo es una colección finita, homogénea y ordenada de elementos. Finita: Todo arreglo tiene un límite; es decir, debe determinarse cuál será el número máximo de elementos que podrán formar parte del arreglo. Homogénea: Todos los elementos del arreglo deben ser del mismo tipo. Ordenada: Se puede determinar cuál es el primer elemento, el segundo, el tercero,.... y el n-ésimo elemento. Los arreglos se clasifican de acuerdo con el número de dimensiones que tienen. Así se tienen los:

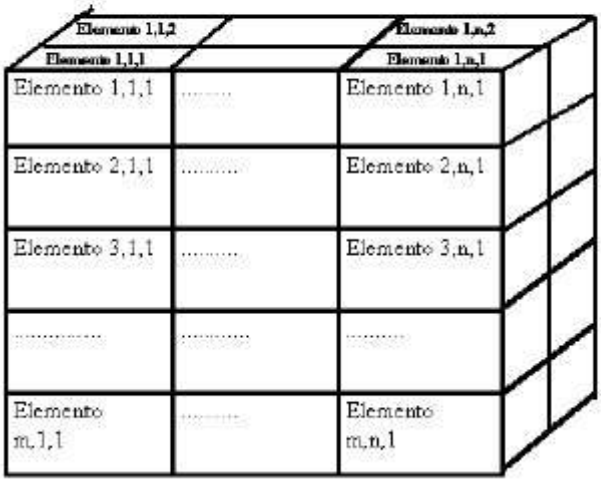
- Unidimensionales (vectores).
- Bidimensionales (tablas o matrices).
- Multidimensionales (tres o más dimensiones).

Elemento 1
Elemento 2
Elemento 3
.....
Elemento n

Elemento 1,1	.....	Elemento 1,n
Elemento 2,1	.....	Elemento 2,n
Elemento 3,1	.....	Elemento 3,n
.....	.....	.....
Elemento m,1	.....	Elemento m,n

Array Unidimensional

Array Bidimensional



Array Multidimensional

PROBLEMA.
Suponga que se desea desarrollar un programa para:

1. Leer una lista de calificaciones de un examen
2.Encontrar su media
3.Escribir una lista de las calificaciones mayores que la media
4.Ordenar la lista de las calificaciones en orden ascendente.

Supongamos también que hay 100 calificaciones. Debemos utilizar 100 variables diferentes nota1, nota2, ..., nota100, de ese modo son 100 direcciones diferentes de memoria para almacenar las calificaciones del examen. Se imagina declarar las 100 variables, ¿cuántas instrucciones involucra?

nota1, nota2, nota3,.....nota100

(En la declaración real de un programa no pueden usarse puntos suspensivos, por lo tanto serán 100 veces). En la fase de lectura de datos, serán también 100 veces las instrucciones para ir leyendo cada valor. Leer (nota1, nota2, nota3,....., nota100)

Para calcular la media:
Media = (nota1+nota2+.....+nota100)/100
Para la lista de calificaciones mayores que la media, deberá también irse comparando una por una:

```
Si nota1 > media entonces
  Imprimir nota1
Fin-si

Si nota2 > media entonces
  escribir nota2
Fin-si

Si nota100 > media entonces
  escribir nota100
Fin-si
```

Y después de más de 450 líneas de código..... ¡Falta ordenar la lista de calificaciones en orden ascendente!

Después que aprendas a usar arreglos verás cómo se ahorra instrucciones porque es fácil recorrer toda la lista de notas con unas pocas instrucciones. En el caso anterior, cuando el acceso a la información es secuencial, sólo se puede acceder a un elemento buscando desde el principio de la lista, y esto es algo lento. Lo que se necesita es una estructura de acceso directo que permita almacenar y recuperar los datos directamente especificando su posición en la estructura, de esa manera se requerirá el mismo tiempo para acceder al elemento de la posición 100 que el de la posición 5.

También preferiremos que esta estructura se almacene en memoria principal para que su almacenaje y recuperación sea más rápida. Es por ello que existen los arreglos, que están organizados en una secuencia de elementos, todos del mismo tipo y se puede acceder a cada elemento directamente especificando su posición en esta secuencia.

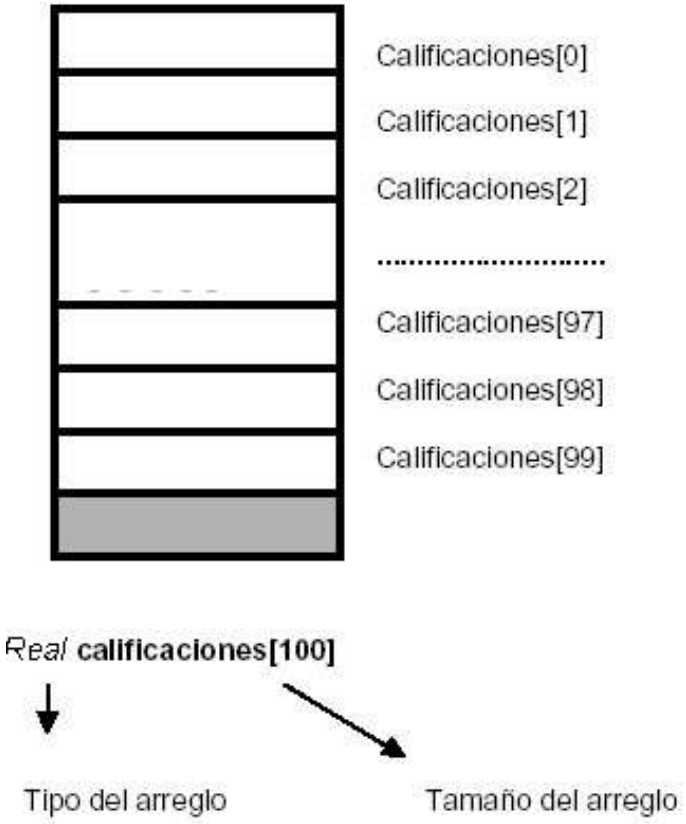
6.1 Arreglos Unidimensionales

Están formados por un conjunto de elementos de un mismo tipo de datos que se almacenan bajo un mismo nombre, y se diferencian por la posición que tiene cada elemento dentro del arreglo de datos. Al declarar un arreglo, se debe inicializar sus elementos antes de utilizarlos. Para declarar un arreglo tiene que indicar su tipo, un nombre único y la cantidad de elementos que va a contener. Por ejemplo, las siguientes instrucciones declaran tres arreglos distintos:

costoPartes[50]

edadEmpleados[100]

nombres [60]



Para acceder a valores específicos del arreglo, use un valor de índice que apunte al elemento deseado. Por ejemplo, para acceder al primer elemento del arreglo calificaciones debe utilizar el valor de índice 0 (calificaciones[0]). Los programas en C++ siempre indizan el primer elemento de un arreglo con 0 y el último con un valor menor en una unidad al tamaño del arreglo.

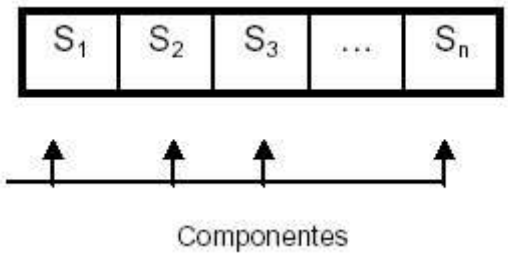
Para inicializar todos los elementos de una vez, se colocan dentro de una estructura for que va del primer elemento al último que contiene el arreglo. Para asignar un valor a un elemento del arreglo se hace por ejemplo:

Calificaciones[0] = 100

Cuando se usan arreglos, una operación común es usar una variable índice para acceder a los elementos de un arreglo. Suponiendo que la variable índice i contiene el valor 3, la siguiente instrucción asigna el valor 400 a valores[3]:

valores[i] = 400 <-> valores[3] = 400

Partes de un arreglo:
Los componentes. Hacen referencia a los elementos que forman el arreglo, es decir, a los valores que se almacenan en cada una de las casillas del mismo. Los índices. Permiten hacer referencia a los componentes del arreglo en forma individual, especifican cuántos elementos tendrá el arreglo y además, de qué modo podrán accesarse esos componentes.



Definición de Arreglos

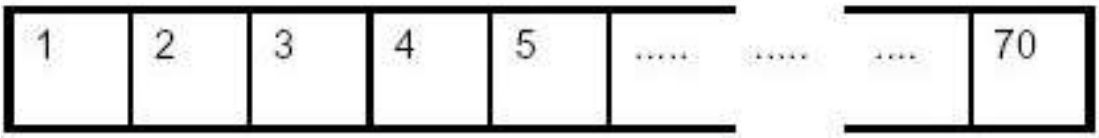
identArreglo [Limsup];

Operaciones con Vectores:
Las operaciones que se pueden realizar con vectores durante el proceso de resolución de un problema son:

- Lectura/ escritura
- Asignación
- Actualización (inserción, eliminación, modificación)
- Recorrido (acceso secuencial)
- Ordenación
- Búsqueda

Ejemplos:
Sea arre un arreglo de 70 elementos enteros con índices enteros. Su representación nos queda:

arre[70];



Lectura

El proceso de lectura de un arreglo consiste en leer y asignar un valor a cada uno de sus elementos. Normalmente se realizan con estructuras repetitivas, aunque pueden usarse estructuras selectivas. Usamos los índices para recorrer los elementos del arreglo:

```
Hacer para i = 0 a 69
  Leer arre[i]
Fin-para
```

Escritura

Es similar al caso de lectura, sólo que en vez de leer el componente del arreglo, lo escribimos.

```
Entero arre[n], n

Inicio

        Escriba "Digite el tamaño del arreglo"

        Leer n
        Hacer para i = 0 a n -1
           Imprimir arre[i]
        Fin-para

Fin
```

Asignación

No es posible asignar directamente un valor a todo el arreglo; sino que se debe asignar el valor deseado en cada componente. Con una estructura repetitiva se puede asignar un valor a todos los elementos del vector.
Por ejemplo:

arre[1] = 120 (asignación de un valor constante único a una casilla del vector)
arre[3] = arre[1] / 4 (asignar una operación)

Se puede asignar un valor constante a todos los elementos del vector:

```
Hacer para i = 0 a 4
   arre[i] = 3
Fin-para
```

Acceso Secuencial. (Recorrido)

El acceso a los elementos de un vector puede ser para leer en él o para escribir (visualizar su contenido). Recorrido del vector es la acción de efectuar una acción general sobre todos los elementos de ese vector.

Actualización

Incluye añadir (insertar), borrar o modificar algunos de los ya existentes. Se debe tener en cuenta si el arreglo está o no ordenado. Añadir datos a un vector consiste en agregar un nuevo elemento al final del vector, siempre que haya espacio en memoria.

6.1.1 PROBLEMAS PROPUESTOS

1) Almacenar en un vector los números de 1 hasta 100 e indicar los números pares.

```
Entero arreglo[100], control

Inicio
        Hacer para control = 0 hasta 99
                arreglo[control] = control + 1
        Fin para
        Hacer para control = 0 hasta 99
                Si arreglo[control] Mod 2 = 0 entonces
                        Escribir "El número " , arreglo[control] , " es par."
                Fin si
                Escribir arreglo[control]
        Fin para
        arreglo[5] = 20
        Escribir arreglo[40]
        Escribir[14]
Fin
```

2) Almacenar en un vector los números de 1 hasta n.

3) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir su contenido.

4) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir los números pares e impares.

5) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir los números primos.

6) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir los números menores a 500.

7) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir los números primos.

8) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir los números primos y la cantidad de números que no son primos.

9) Almacenar en un vector los números de 1 hasta n y posteriormente imprimir la sumatoria de los números primos y no primos.

10) Escribir un programa solicite al usuario 100 valores enteros y los almacene en un vector. Posteriormente, debe calcular el valor máximo, mínimo y la media

11) Calcular el promedio de 50 valores almacenados en un vector. Determinar además cuantos son mayores que el promedio, imprimir el promedio, el número de datos mayores que el promedio y una lista de valores mayores que el promedio.

12) Escribir un programa que lea un vector de números enteros (de una sola cifra) y compruebe si el número formado por cada uno de los elementos del vector es capicúa o no.

13) Capturar 4 sueldos y desplegarlos aumentados en un 25%

14) Llenar dos vectores A y B de 45 elementos cada uno, sumar el elemento uno del vector A con el elemento uno del vector B y así sucesivamente hasta 45, almacenar el resultado en un vector C, e imprimir el vector resultante.

15) Las claves de respuesta de 5 preguntas están guardadas en un arreglo:

res[0]="A";

res[1]="C";

res[2]="A";

res[3]="B";

res[4]="A";

y las respuestas del usuario deben almacenaren en otro arreglo, indicar cuantos respuestas fueron contestadas correctamente, incorrectamente e indicar si pasó la prueba donde 3 es la cantidad de preguntas mínimas que debe contestar bien para pasar la prueba.

1. Requerimientos

- Realizar la revisión de una prueba o examen de 5 preguntas de una persona.
- Las respuestas correctas de las 5 preguntas son:

  res[0]="A";

  res[1]="C";

  res[2]="A";

  res[3]="B";

  res[4]="A";

- Las respuestas del usuario deben ser almacenada en otro arreglo.
- Indicar cuantas respuestas fueron correctas e incorrectas.
- La prueba es aprobada con tres (3) o más respuestas correctas.
- Indicar si el usuario aprobó o no la prueba.

2. Análisis

- Definir un arreglo de cinco (5) índices con las respuestas correctas a las cinco (5) preguntas.
- Definir un arreglo de cinco (5) índices para almacenar las respuestas del usuario.
- Definir un contador para conocer la cantidad de respuestas correctas.
- Definir un contador para conocer la cantidad de respuestas incorrectas.
- Realizar un ciclo de cinco iteraciones para almacenar las respuestas del usuario.

REPUESTAS CORRECTAS	
ÍNDICE	VALOR
0	A
1	C
2	A
3	B
4	A

REPUESTAS USUARIO	
ÍNDICE	VALOR
0	A
1	B
2	B
3	C
4	A

- Realizar un ciclo de 5 iteraciones que me permita recorrer el contenido de los dos arreglos (respuestas correctas, respuestas usuario).
- Debo realizar una condición donde evalué el contenido de un número de índice de los dos (2) arreglos (respuestas correctas, respuestas usuario).
- En caso de que la respuesta del arreglo de correctas con respecto del arreglo de usuario sean iguales incremento el contador de correctas en uno (1).
- En caso de que la respuesta del arreglo de correctas con respecto del arreglo de usuario NO sean iguales incremento el contador de incorrectas en uno (1).
- Después de salir del ciclo evalúo si el contador de respuestas correctas es mayor o igual 3 para informarle al usuario si aprobó o No.
- Imprimir la cantidad de respuestas correctas e inconrrectas.

TIPO	VARIABLE	DESCRIPCIÓN	OPERACIÓN
ARREGLO	respuCorre	almacena las 5 respuestas correctas	Se alimentan directamente en le codigo fuente
ARREGLO	respuUsuar	almacena las 5 respuestas del usuario	Solicitado
ENTERO	contaCorre	almacena la cantidad de respuestas del usuario correctas	contaCorre++
ENTERO	contaIncor	almacena la cantidad de respuestas del usuario incorrectas	contaIncor++
ENTERO	control	valida la cantidad de iteraciones de un ciclo for	contro = 0; control < 4; control++

3. Diseño

4. Codificación

5. Pruebas

16) Llenar un vector de 20 elementos, imprimir la posición y el valor del elemento mayor almacenado en el vector. Suponga que todos los elementos del vector son diferentes.

17) Almacenar 500 números en un vector, elevar al cuadrado cada valor almacenado en el vector, almacenar el resultado en otro vector. Imprimir el vector original y el vector resultante.

18) Almacenar 300 números en un vector, imprimir cuantos son ceros, cuántos son negativos, cuantos positivos. Imprimir además la suma de los negativos y la suma de los positivos.

19) Almacenar 150 números en un vector, almacenarlos en otro vector en orden inverso al vector original e imprimir el vector resultante.

20) Se tienen almacenados en la memoria dos vectores M y N de cien elementos cada uno. Hacer un algoritmo que escriba la palabra “Iguales” si ambos vectores son iguales y “Diferentes” si no lo son.

Serán iguales cuando en la misma posición de ambos vectores se tenga el mismo valor para todos los elementos.

21) Se tiene el vector A con 100 elementos almacenados. Diseñe un algoritmo que escriba “SI” si el vector esta ordenado ascendentemente o “NO” si el vector no está ordenado.

22) Diseñe un algoritmo que lea un número cualquiera y lo busque en el vector X, el cual tiene almacenados 80 elementos. Escribir la posición donde se encuentra almacenado el número en el vector o el mensaje “NO” si no lo encuentra. Búsqueda secuencial.

23) Diseñe un algoritmo que lea dos vectores A y B de 20 elementos cada uno y multiplique el primer elemento de A con el último elemento de B y luego el segundo elemento de A por el diecinueveavo elemento de B y así sucesivamente hasta llegar al veinteavo elemento de A por el primer elemento de B. El resultado de la multiplicación almacenarlo en un vector C.

24) Diseñe un algoritmo que almacene en un vector llamado FIB[100] los 100 primeros números de la serie Fibonacci.

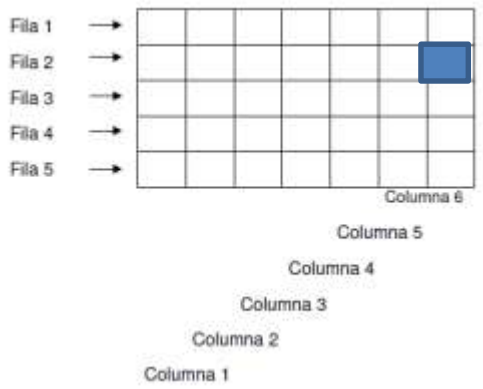
25) Leer una lista de n valores reales y calcular la media y obtener la desviación de cada uno de los valores respecto a la media (desviación = x[i] - media). El número máximo de valores a introducir en 100.

26) Se define un vector de enteros como alternado si el valor de cada componente es de signo contrario a la componente anterior. Escribir un programa que lea desde teclado un vector de enteros, compruebe si es alternado, y finalmente imprima por pantalla un mensaje indicando si es alternado o no. Un ejemplo de vector alternado:

-2	3	-4	1	-90	2

6.2. Arreglo Bidimensional

Es un conjunto de datos homogéneo, finito y ordenado, donde se hace referencia a cada elemento por medio de dos índices. El primero se utiliza para los renglones (filas) y el segundo para las columnas. También puede definirse como un arreglo de arreglos. Internamente en memoria se reservan MxN posiciones consecutivas para almacenar todos los elementos del arreglo.



Declaración de una matriz:

idArreglo[lim_sup_row, lim_sup_col]

El recorrido por columnas se hace de manera similar, invirtiendo el sentido de los índices.

```
Hacer para j = 0 a (n – 1)
    Hacer para i = 0 a (m -1)
        Imprimir a[i,j]
    Fin-para
Fin-para
```

Ejemplos.

1) Rellenar una matriz identidad de 4 por 4 elementos.
Una matriz identidad es aquella en la que la diagonal principal está llena de unos y el resto de los elementos son cero. Para llenar la matriz identidad se debe verificar que cuando los índices i y j sean iguales, la posición vale 1, en caso contrario se asigna cero al elemento i,j.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

```
Algoritmo
Inicio

    Matriz [3,4]
    Hacer para i = 0 a 2
        Hacer para j = 0 a 3
            Si i=j entonces
                Matriz[i,j] =1
            Si-no
                Matriz[i,j] =0
            Fin-si
        Fin-para
    Fin-para
Fin
```

6.2.1 PROBLEMAS PROPUESTOS

1. Hacer un algoritmo que almacene números en una matriz de 5 * 6. Imprimir la suma de los números almacenados en la matriz.

1. Requerimientos

- Almacenar números en un arreglo de 5*6
- Imprimir la sumatoria del contenido del arreglo.

2. Análisis

- Definimos un arreglo de 5*6.
- Definir una variable que almacene la sumatoria de los números que contiene el arreglo.
- Creamos un ciclo para con el fin de recorrer las filas del arreglo.
- Creamos un ciclo para con el fin recorrer las columnas del arreglo.
- Solicitamos los números a almacenar en el arreglo.
- Almacenamos los valores ingresados por el usuario al arreglo
- Creamos un ciclo para con el fin de recorrer las filas del arreglo.
- Creamos un ciclo para con el fin recorrer las columnas del arreglo.
- Obtenemos los valores que contiene el arreglo.
- Acumulamos los valores obtenidos del arreglo.
- Después de finalizar el ciclo imprimo el valor de la sumatoria de los números contenidos en el arreglo.

TIPO	VARIABLE	DESCRIPCIÓN	OPERACIÓN
ARREGLO	numeros[5,6]	almacena los números ingresados por el usuario.	numeros[i,j]=
entero	sumatoria	almacena la sumatoria de los números ingresados por el usuari	sumatoria = sumatoria + numeros[i,j]
entero	i	variable de control del primer ciclo para	i = 0; i < 5; i++
entero	j	variable de control del segundo ciclo para	j = 0; j < 6; j++

3. Diseño

4. Codificación

```
Entero i, j, sumatoria
Arreglo numeros[5,6]
Inicio
        sumatoria = 0
        Para i =0 hasta 4 hacer
                Para j =0 hasta 5 hacer
                        Escriba "Por favor ingrese un número"
                        Lea numeros[i,j]
                Fin para
        Fin para
        Para i=0 hasta 4 hacer
                Para j=0 hasta 5 hacer
                        sumatoria  = sumatoria  + numeros[i,j]
                Fin para
        Fin para
        Escriba "La sumatoria del contenido del arreglo es: " , sumatoria
Fin
```

5. Pruebas

i	j		sumatoria	numeros	0	1	2	3
	0	0	0	0	4	8	9	3
	1	1	4	1	3	89	6	76
	2	2	12	2	23	45	90	5
	3	3	21	3	2	4	5	8
	4	4	24	4	45	67	34	23
	5	5	29					
	0	6	96					
	1	0 .						
		1 .						
		2 .						
		3	589					
		4						
		5						
		6						
		0						
		1						
		2						
		3						
		4						
		5						
		6						
		0						
		1						
		2						
		3						
		4						
		5						
		6						
		0						
		1						
		2						
		3						
		4						
		5						
		6						

2. Hacer un algoritmo que llene una matriz de 10 * 10 y determine la posición [renglón, columna] del número mayor almacenado en la matriz. Los números son diferentes.

3. Hacer un algoritmo que llene una matriz de 7 * 7. Calcular la suma de cada renglón y almacenarla en un vector, la suma de cada columna y almacenarla en otro vector.

4. Hacer un algoritmo que llene una matriz de 20 * 20. Sumar las columnas e imprimir que columna tuvo la máxima suma y la suma de esa columna.

5. Hacer un algoritmo que llene una matriz de 5 * 5 y que almacene la diagonal principal en un vector. Imprimir el vector resultante.

6. Hacer un algoritmo que llene una matriz de 10 * 10 y que almacene en la diagonal principal unos y en las demás posiciones ceros.

7. Hacer un algoritmo que llene una matriz de 6 * 8 y que almacene toda la matriz en un vector. Imprimir el vector resultante.

8. Hacer un algoritmo que llene una matriz de 8 * 8, que almacene la suma de los renglones y la suma de las columnas en un vector. Imprimir el vector resultante.

9. Hacer un algoritmo que llene una matriz de 5 * 6 y que imprima cuantos de los números almacenados son ceros, cuántos son positivos y cuantos son negativos.

10. Diseñe un pseudocódigo que escriba el número de la hilera cuya suma sea mayor que las demás hileras. Suponga que todas las hileras suman diferente cantidad.

11. El dueño de una cadena de tiendas de artículos deportivos desea controlar sus ventas por medio de una computadora. Los datos de entrada son :

a) El número de la tienda (1 a 50)
b) Un número que indica el deporte del articulo (1 a 20)
c) El costo del artículo.

Hacer un pseudocódigo que escriba al final del día lo siguiente

- Las ventas totales en el día para cada tienda
- Las ventas totales para cada uno de los deportes.
- Las ventas totales de todas las tiendas.

12. El departamento de policía de la ciudad de Tuxtepec ha acumulado información referente a las infracciones de los límites de velocidad durante un determinado periodo de tiempo. El departamento ha dividido la ciudad en cuatro cuadrantes y desea realizar una estadística de las infracciones a los límites de velocidad en cada uno de ellos. Para cada infracción se ha preparado una tarjeta que contiene la siguiente información:

- número de registro del vehículo;
- cuadrante en el que se produjo la infracción
- límite de velocidad en milla por hora

- Diseñe un diagrama para producir 2 informes; el 1o. Que contiene una lista de la multa de velocidad recolectada, donde la multa se calcula como la suma del costo de la corte ($20,000) mas $ 1,250 por cada mph que exceda la velocidad limite. Prepare una tabla con los siguientes resultados:

INFRACCIONES A LOS LIMITES DE VELOCIDAD

Registro del	Velocidad	Velocidad	Multa

Este informe debe ser seguido de un segundo en el cual se proporcione un análisis de las infracciones por cuadrante. Para cada uno de los 4 cuadrantes mencionados, debe darse el número de infracciones y la multa promedio.

13. Se tiene almacenada la matriz M (50,5) la cual contiene la información sobre las calificaciones de la materia de LENGUAJES ALGORITMICOS. Diseñe un pseudocódigo que imprima:

- Cantidad de alumnos que aprobaron la materia.
- Cantidad de alumnos que tienen derecho a nivelación.
- El (o los) numero (s) de control de lo(s) alumno(s) que haya (n) obtenido la máxima calificación final

VII FUNCIONES

Habíamos visto la programación estructurada que permite la escritura de programas fáciles de leer y modificar. En esta programación, el flujo lógico se gobierna por las estructuras de control básicas vista hasta hoy: secuenciales, repetitivas y de selección. La programación modular permite la descomposición de un problema en un conjunto de subproblemas independientes entre sí, más sencillos de resolver y que pueden ser tratados separadamente unos de otros. Gracias a la modularidad se pueden probar los subprogramas o módulos de manera independiente, depurándose sus errores antes de su inclusión en el programa principal y almacenarse para su posterior utilización cuantas veces se precise.

7.1 Módulo

Uno de los elementos principales de programación utilizados en la representación de cada módulo es la subrutina. Una subrutina es un conjunto de instrucciones de cómputo que realizan una tarea. Un programa principal llama a estos módulos a medida que se necesitan. Un módulo es un segmento, rutina, subrutina, sub-algoritmo o procedimiento, que puede definirse dentro de un algoritmo con el fin de ejecutar una tarea específica y puede ser llamada o invocada desde el algoritmo principal cuando sea necesario. Los módulos son independientes en el sentido de que ningún módulo puede tener acceso directo a cualquier otro módulo, con excepción del módulo al que llama y sus propios submódulos. Sin embargo, los resultados producidos por un módulo pueden ser utilizados por cualquier otro módulo cuando se transfiera a ellos el control. Los módulos tienen una entrada y una salida. Se pueden tomar decisiones dentro de un módulo que tenga repercusión en todo el flujo, pero el salto debe ser únicamente hacia el programa principal. Al descomponer un programa en módulos independientes más simples se conoce también como el método de "Divide y vencerás".
¿Cuándo es útil la modularización?
Este enfoque de segmentación o modularización es útil en dos casos:
1. Cuando existe un grupo de instrucciones o una tarea específica que deba ejecutarse en más de una ocasión.
2. Cuando un problema es complejo o extenso, la solución se divide o segmenta en módulos que ejecutan partes o tareas específicas.

7.1.1 Ventajas de la Programación Modular

Como los módulos son independientes, el desarrollo de un programa se puede efectuar con mayor facilidad, ya que cada módulo se puede crear aisladamente y varios programadores podrán trabajar simultáneamente en la confección de un algoritmo, repartiéndose las distintas partes del mismo. Se podrá modificar un módulo sin afectar a los demas Las tareas, sub-algoritmos, sólo se escribirán una vez, aunque se necesiten en distintas ocasiones a lo largo del algoritmo. El uso de módulos facilita la proyección y la comprensión de la lógica subyacente para el programador y el usuario. Aumenta la facilidad de depuración y búsqueda de errores en un programa ya que éstos se pueden aislar fácilmente. El mantenimiento y la modificación de la programación se facilitan. Los módulos reciben diferentes nombres:

- Funciones en C, C++
- Subrutinas en Basic
- Procedimientos y funciones en Pascal
- Subrutinas en Fortran y
- Secciones en Cobol.

7.1.2 Desarrollar programas de forma modular

Significa que pueden identificarse las principales tareas a realizar por el programa y que se pueden diseñar y probar procedimientos individuales para estas tareas. Por ejemplo: ¿Qué transacciones se le hacen a una cuenta de ahorros? Transacciones:

- depósito (cheque y efectivo)
- intereses
- retiro
- estado de cuenta
- cambio de libreta

7.1.3 Tiempo de vida de los datos – Ámbito.

Según el lugar donde son declaradas puede haber dos tipos de variables.

G*lobales*: las variables permanecen activas durante todo el programa. Se crean al iniciarse éste y se destruyen de la memoria al finalizar. Pueden ser utilizadas en cualquier procedimiento o función.

***Locales*:** las variables son creadas cuando el programa llega a la función o procedimiento en la que están definidas. Al finalizar la función o el procedimiento, desaparecen de la memoria. Si dos variables, una global y una local, tienen el mismo nombre, la local prevalecerá sobre la global dentro del módulo en que ha sido declarada. Dos variables locales pueden tener el mismo nombre siempre que estén declaradas en funciones o procedimientos diferentes.

Parámetros Formales

Es un tipo especial de variables en un procedimiento a los que se pueden pasar valores desde el exterior del procedimiento. Se declaran en la cabecera del procedimiento.

Ejemplos:

1. Uso de variables globales en procedimientos o funciones.

```
Inicio
      Entero x <-20;
       Cambiar();
       Escribir (x)
fin


Modulo Cambiar()
Inicio

          Escribir x
          x <-1;
Fin
```

La variable x está definida como global, por lo tanto la salida será 1.

2. Uso de variables locales.

```
Algoritmo local

Inicio
    Entero x <-0;
    cambiar();
    escribir (x);
fin

Modulo cambiar()
inicio
        Entero x <-1;
fin

Fin-modulo
```

Como x es local, no tiene efecto en el programa, por lo tanto la salida será 0.

7.1.4 PARÁMETROS

Un parámetro es un tipo de variable que es recibida por una función, procedimiento o subrutina.

7.1.4.1 Parámetro Por valor

Significa que la función (o subrutina) recibe sólo una copia del valor que tiene la variable, o sea que no la puede modificar.

Ejemplo:

```
Algoritmo parámetro valor

Inicio
    Entero x <- 0;
    Cambiar(x);
```

```
        Escribir(X);
Fin


Modulo Cambiar (Entero y)
inicio
       y <-1;
fin

Fin-modulo
```

Salida 0

7.1.4.2 Parámetro Por referencia

Significa que se pasa la posición de memoria donde esta guardada la variable, por lo que la función puede saber cuánto vale, pero además puede modificarla de cualquier manera.G

Ejemplo:
Algoritmo parámetro referencia.

```
Inicio
       Entero x <-0;
       Cambiar (x);
       Escribir (x);
Fin


Modulo Cambiar (Ref Entero  y)
Inicio
   y <-1;
Fin

Fin-modulo
```

Salida: x =1

7.1.5 FUNCIONES

La función es una estructura autónoma similar a los módulos. La diferencia radica en que la función se usa para devolver un solo valor de un tipo de dato simple a su punto de referencia. La función se relaciona especificando su nombre en una expresión, como si fuera una variable ordinaria de tipo simple. Las funciones se dividen en estándares y definidas por el usuario.

- **Estándar**: Son funciones proporcionadas por cualquier lenguaje de programación de alto nivel, y se dividen en aritméticas y alfabéticas.

- Definidas por el usuario: son funciones que puede definirlas el programador con el propósito de ejecutar alguna función específica, y que por lo general se usan cuando se trata de hacer algún cálculo que será requerido en varias ocasiones en la parte principal del algoritmo.

Ejemplos:

```
Entero Funcion factorial (Entero n)

inicio

        Entero i, factorial;

        Si n <=1 Entonces
                factorial <-- 1;
        Si-no
                factorial <-- 1;
                Para i = 1 hasta n hacer
                        factorial <-- factorial * 1;
                Fin-para
        Fin-si

        Retornar factorial;

Fin

Fin-funcion
```

Evaluar la función f = x! / (y!(x-y)!)

```
Algoritmo hallar f

Inicio

    Entero x,y;

    Real f;
    Leer x,y;
    f <-- factorial (x)/(factorial (y)* factorial (x-y));
    Escribir "El valor de f es:", f;
Fin
```

7.1.6 PROBLEMAS PROPUESTOS

1) Realizar una calculadora.

2) Solicitar 2 números. Hallar la sumatoria de los números primos de cada uno de ellos y finalmente multiplicar las dos sumatorias y el resultado elevarlo al cuadrado.

VIII METODOS DE ORDENAMIENTO

8. 1 ORDENAMIENTOS INTERNOS Y EXTERNOS